LUCÉLIA DE SOUZA

UM MODELO DE PROVENIÊNCIA PARA EXTRAÇÃO DE TENDÊNCIAS EM SÉRIES TEMPORAIS

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.
Orientador: Prof. Dr. Marcos Sfair Sunye
Coorientadora: Profa. Dra. Maria Salete Marcon Gomes Vaz

CURITIBA

2014

LUCÉLIA DE SOUZA

UM MODELO DE PROVENIÊNCIA PARA EXTRAÇÃO DE TENDÊNCIAS EM SÉRIES TEMPORAIS

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.
Orientador: Prof. Dr. Marcos Sfair Sunye
Coorientadora: Profa. Dra. Maria Salete Marcon Gomes Vaz

CURITIBA

2014

S729m Souza, Lucélia de
Um modelo de proveniência para extração de tendências em séries temporais / Lucélia de Souza. – Curitiba, 2014.
256f. : il. [algumas color.] ; 30 cm.

Tese (doutorado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-graduação em Informática, 2014.
Orientador: Marcos Sfair Sunye -- Coorientadora: Maria Salete Marcon Gomes Vaz.
Bibliografia: p. 201-216.

1. Análise de séries temporais – Processamento de dados. 2. Ontologias (Recuperação da informação). I. Universidade Federal do Paraná. II. Sunye, Marcos Sfair. III. Vaz, Maria Salete Marcon Gomes. IV. Título.

CDD: 519.55

Ministério da Educação
Universidade Federal do Paraná
Programa de Pós-Graduação em Informática

PARECER

Nós, abaixo assinados, membros da Banca Examinadora da defesa da aluna de Doutorado em Ciência da Computação, Lucélia de Souza, avaliamos a tese de doutorado intitulada *"Um Modelo de Proveniência para Extração de Tendências em Séries Temporais"*, cuja defesa pública foi realizada no dia 29 de agosto de 2014, às 15:30 horas, no Departamento de Informática do Setor de Ciências Exatas da Universidade Federal do Paraná. Após avaliação, decidimos pela:

(X) **aprovação** da candidata. ( ) **reprovação** da candidata.

Curitiba, 29 de agosto de 2014.

Prof. Dr. Marcos Sfair Sunye
DINF/UFPR – Orientador

Profa. Dra. Maria Salete Marcon Gomes Vaz
UEPG – Coorientadora

Prof. Dr. César Augusto Tacla
UTFPR – Membro Externo

Prof. Dr. José Carlos Ferreira Da Rocha
UEPG – Membro Externo

Profa. Dra. Angelita Maria De Ré
UNICENTRO – Membro Externo

Prof. Dr. Marcos Didonet Del Fabro
DINF/UFPR – Membro Interno

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
Setor de Ciências Exatas
Coordenação de Pós-Graduação em Informática

AGRADECIMENTOS

Agradeço à Deus pela força de todos os dias e pelo dom da vida.

Agradeço a todos os professores e pesquisadores que diretamente ou indiretamente contribuíram para o desenvolvimento desta tese e aos professores orientadores, prof. Dr. Marcos Sfair Sunye e profa. Dra. Maria Salete Marcon Gomes Vaz.

Agradeço todo o apoio da Universidade Estadual do Centro-Oeste - UNICENTRO e do Departamento de Ciência da Computação, assim como da Universidade Federal do Paraná - UFPR e da Universidade Estadual de Ponta Grossa - UEPG.

RESUMO

Muitas áreas do conhecimento estão relacionadas com a análise de séries temporais, as quais são constituídas por uma sequência de observações de dados sobre o tempo. A análise de séries temporais difere da análise de dados tradicional, dada sua natureza intrínseca, onde as observações são dependentes. Nesse caso, procedimentos estatísticos considerando a independência dos dados não se aplicam, sendo necessário o uso de métodos específicos. Geralmente, a análise de séries temporais ocorre em duas fases, pré-processamento e análise dos dados. Na fase de pré-processamento, são feitas correções para remoção de fenômenos que ocorrem ao longo do tempo, como a extração de tendências (*detrending*). Vários softwares de *detrending* podem ser aplicados para esse fim, melhorando a análise, assim como a maioria dos métodos estatísticos são desenvolvidos para séries temporais estacionárias. Em um processo de *detrending*, informações de proveniência sobre as séries temporais e como as mesmas foram corrigidas de tendências nem sempre são explícitas e de fácil interpretação. Tais informações podem ser obtidas pelo uso de metadados, os quais podem gerar ambiguidades nos resultados gerados, assim como podem ser insuficientes para semanticamente enriquecer o processo de *detrending*. Por outro lado, ontologias permitem gerar e compartilhar conhecimento sobre as séries temporais e sobre os métodos estatísticos aplicados para sua correção, assim como permitem inferências. O principal objetivo desta tese é definir um modelo de proveniência usando ontologias para enriquecer semanticamente a extração de tendências em séries temporais. O modelo é validado por um estudo de caso com séries temporais fotométricas reais. A principal contribuição é a geração de conhecimento semântico, permitindo identificar, além dos dados, agentes e processos envolvidos, informações quanto aos métodos estatísticos usados para *detrending*, facilitando o entendimento de como as séries temporais foram geradas e corrigidas, melhorando a tomada de decisão quanto ao uso de métodos estatísticos. O ineditismo desta tese é a definição de um modelo de proveniência para extração de tendências, apresentando um projeto modular, centrado no reuso e na extensão de ontologias para gerar proveniência sobre séries temporais e processos de *detrending*, enriquecendo semanticamente um passo relevante da fase de pré-processamento da análise de séries temporais, contribuindo para a geração do conhecimento científico.

Palavras-chave: Modelo de Proveniência, Ontologias, OWL, Séries Temporais Não-Estacionárias, Extração de Tendências

ABSTRACT

Nowadays, many knowledge areas are related with the time series analysis, which are constituted by a sequence of data observation at the time. The time series analysis is different from the traditional data analysis, due to their intrinsic nature, where the observations are dependent. In this case, statistical procedures considering the data′s independence are not applied, being necessary the use of specific methods. Usually, the time series analysis occurs in two phases, preprocessing and data analysis. In the preprocessing phase, corrections are done to remove phenomena that occur throughout the time, like the trend extraction (detrending). Many detrending software can be applied for this objective, improving the analysis, as well as the most of statistical methods are developed to stationary time series. In a detrending process, provenance information about the time series and how the time series were detrended are not always explicit and easy to interpret. Such information can be obtained by metadata, which can generate ambiguity in the results generated and they can also be insufficient to semantically enrich the detrending process. On the other hand, ontologies allow generating and sharing knowledge about the time series and on the statistical methods used for it′s correction, as well as allow inferences. The main goal of this doctoral thesis is to define a provenance model using ontologies to semantically enrich the trend extraction of time series. The model is validated by a case study involving real photometric time series. The main contribution is the semantic knowledge generation, allowing to identify, besides the data, agents and process involved, information about the statistical methods used for detrending, facilitating the understanding about how the time series were generated and corrected, improving the decision making related with the statistical methods applicability. The novelty of this doctoral thesis is the definition of a provenance model for trend extraction, presenting a modular design, centered on reuse and on the ontologies extension to generate provenance about time series and detrending processes, enriching semantically a relevant step of preprocessing phase of the time series analysis, contributing to the generation of the scientific knowledge.

Keywords: Provenance Model, Ontologies, OWL, Nonstationary Time Series, Detrending

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ACM - *Association for Computing Machinery*

CDA - *Corot Detrend Algorithm*

CDA-M - *Corot Detrend Algorithm Modified*

DO - *Detrend Ontology*

DPM - *Detrend Provenance Model*

DUL - *Dolce Ultralite*

EEMD - *Ensemble Empirical Mode Decomposition*

EMD - *Empirical Mode Decomposition*

EOF - *Empirical Orthogonal Function*

FIR - *Finite Impulse Response*

FOL - *First Order Logic*

GOF - *Goodness Of Fit*

HTTP - *HyperText Transfer Protocol*

IIR - *Infinite Impulse Response*

IMFs - *Intrinsic Mode Functions*

LOD - *Linked Open Data*

O&M - *Observations and Measurements*

OGC - *Open Geospatial Consortium*

OLS - *Ordinary Least Squares*

OPM - *Open Provenance Model*

OPMO - *Open Provenance Model Ontology*

OWL - *Ontology Web Language*

PCA - *Principal Component Analysis*

PML - *Proof Markup Language*

RDF - *Resource Description Framework*

RDFS - *Resource Description Framework Scheme*

SPARQL - *SPARQL Protocol and RDF Query Language*

SSA - *Singular Spectrum Analysis*

SQL - *Structured Query Language*

SWE - *Sensor Web Enablement*

SWEET - *Semantic Web for Earth and Environmental Terminology*

SWRL - *Semantic Web Rule Language*

TSO - *Time Series Ontology*

UNA - *Unique Name Assumption*

URI - *Universal Resource Identifier*

URL - *Uniform Resource Locator*

W3C - *World-Wide Web Consortium*

XML - *eXtensible Markup Language*

SUMÁRIO

CAPÍTULO 1

INTRODUÇÃO

1.1 Motivação

Muitas áreas do conhecimento estão relacionadas com a análise de séries temporais [223], a partir das quais são extraídas informações para geração do conhecimento científico. Séries temporais [97] são dados observacionais, geralmente obtidos em intervalos regulares de tempo, gerados a todo instante em ciências físicas, biológicas, sociais, econômicas, ambientais, da saúde, entre outras áreas. Como exemplos de aplicabilidade, na medicina, séries temporais caracterizam dados obtidos a partir de um eletroencefalograma ou um eletrocardiograma; na engenharia, são observados e analisados os movimentos do som, de sinais elétricos e voltagem; na economia, analisa-se a variação dos investimentos na bolsa de ações; na meteorologia, são obtidos dados pluviométricos a partir de sensores, entre outros.

A natureza intrínseca das séries temporais considera que as observações são dependentes ou correlacionadas, onde a ordem das observações é relevante para análise. Ou seja, a análise de um dado, no instante de tempo t, é influenciada pela análise de dados em instantes anteriores. Nesse caso, procedimentos estatísticos e técnicas dependentes da hipótese de independência dos dados não se aplicam, sendo necessário o uso de métodos apropriados. Dessa forma, a análise de séries temporais difere da análise de dados tradicional, onde a maioria dos métodos estatísticos assume que os dados disponíveis podem ser considerados, em alguma instância, como uma amostra aleatória independente, a partir de uma população de interesse [244].

Séries temporais podem apresentar quatro componentes principais [230]: tendência, sazonal, irregular e ciclos. Geralmente, em um processo de análise, esses componentes são decompostos das mesmas. O componente tendência usualmente é definido como uma variação em determinadas medidas das séries temporais, tais como a média, variância ou co-variância e é geralmente caracterizado como um movimento de longo prazo. Sazonal é o componente de variação em uma série temporal, sendo dependente de um determinado período. Cíclico é o componente correspondente às variações cíclicas, cuja periodicidade é desconhecida. Irregular ou Aleatório é o componente que caracteriza os resíduos que sobram quando demais componentes, tais como tendência e sazonalidade, são removidos das séries temporais.

Geralmente, a análise de séries temporais é realizada em duas fases. A primeira é a fase de pré-processamento, seguida pela fase de análise em si [248]. Na fase de pré-processamento, um dos passos considerados mais importantes é a extração de tendências (*detrending*)[1] [97, 59, 248, 193, 183], as quais necessitam ser extraídas porque podem ocultar outros fenômenos, assim como, conforme Bendat e Piersol [70], grandes distorções podem ocorrer em processamentos posteriores da densidade de probabilidade, correlação e quantidades espectrais. Também a maioria dos métodos estatísticos são desenvolvidos considerando a estacionariedade dos dados.

Nesse contexto, vários softwares de extração de tendências, utilizando diferentes métodos estatísticos podem ser usados para correção das séries temporais. A análise das séries

[1]Nessa tese são utilizados ambos os termos, *detrending* e extração de tendências.

temporais pode ser feita usando métodos em seu domínio usual, ou seja, no domínio do tempo ou, no domínio da frequência, onde o componente tendência é analisado sob o sinal inteiro e informações sobre o domínio do tempo são perdidas, havendo também a análise no domínio tempo-frequência, considerando ambos os domínios. Uma das possibilidades para remoção da tendência das séries temporais é fazer uso de métodos estatísticos que permitem sua estimação, a qual é extraída das séries temporais. Dentre os métodos que podem ser utilizados com esse objetivo no domínio do tempo, destacam-se os métodos paramétricos, tais como análise de regressão (linear, não-linear ou múltipla) [144] ou métodos não-paramétricos, tais como regressão não-paramétrica, baseada em alguma forma de suavização das séries temporais [158, 241, 102, 175]. Outros métodos para sua extração incluem o uso de filtros [95, 183].

O conhecimento sobre a proveniência das séries temporais e como um processo de *detrending* foi realizado, contribui para facilitar a tomada de decisão quanto aos dados e procedimentos aplicados, melhorando a análise. Entretanto, o conhecimento sobre a proveniência das séries temporais e a aplicabilidade dos métodos de *detrending* nem sempre é explícito e fácil de interpretar. Nesse contexto, informações de proveniência relacionadas com a origem ou fonte de algo [196] adicionam conhecimento aos dados e processos, permitindo ao pesquisador conhecer melhor as séries temporais, assim como obter informações sobre os métodos estatísticos aplicados para sua correção.

Segundo Tan [234], a proveniência dos dados pode ser obtida por meio de anotações com o uso de metadados, caracterizando dados sobre dados. Entretanto, somente o uso de metadados pode gerar ambiguidades nos resultados gerados, dada sua forma livre de criação, ocasionando falta de interoperabilidade semântica, dificultando seu uso por agentes de software. Nesse sentido, ontologias são usadas para semanticamente enriquecer a geração de informações de proveniência, as quais permitem, não somente representar, mas inferir conhecimento semântico sobre os dados e processos, por meio de raciocínio lógico. Ontologias, segundo Guarino [143], representam uma conceitualização acerca de um vocabulário comum.

A motivação para o desenvolvimento dessa pesquisa está relacionada à geração de informações de proveniência, usando ontologias, para modelagem do passo de extração de tendências, no qual ocorre a transformação das séries temporais não-estacionárias (que apresentam tendências) em estacionárias (livres de tendências). A geração de informações de proveniência quanto a este importante passo evita reprocessamento dos dados e contribui para o compartilhamento, reuso e a geração de novas análises.

O desenvolvimento desta tese contribui para o enriquecimento semântico quanto aos dados, algoritmos e métodos usados em um passo de *detrending*, facilitando o entendimento, por parte dos pesquisadores, sobre qual método foi usado em qual tipo de séries temporais, assim como auxilia na tomada de decisão sobre outros métodos que podem ser aplicados para correção de tendências, objetivando melhores resultados e contribuindo para a geração do conhecimento científico.

O problema de pesquisa desta tese é a geração de conhecimento semântico quanto à extração de tendências em séries temporais. As principais questões identificadas em relação ao problema de pesquisa são:

- "Como gerar conhecimento semântico na aplicabilidade de métodos de *detrending* em séries temporais?"
- "Como permitir interoperabilidade semântica quanto ao uso de métodos estatísticos para extração de tendências em séries temporais?"

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral desta tese é a definição de um modelo de proveniência, baseado em ontologias, para enriquecer semanticamente o passo de extração de tendências em séries temporais, facilitando a tomada de decisão, por parte dos pesquisadores, e contribuindo com a geração do conhecimento científico.

1.2.2 Objetivos Específicos

Como objetivos específicos se destacam:

- Definir uma ontologia de domínio, centrada no reuso, em linguagem *Ontology Language Web - OWL*, contendo informações de proveniência quanto às séries temporais.
- Definir uma ontologia de domínio, centrada no reuso, em linguagem OWL, para descrição dos principais métodos estatísticos com aplicabilidade em um processo de *detrending* no domínio do tempo, permitindo extensibilidade para o domínio da frequência.
- Definir um modelo de proveniência para *detrending*, utilizando as ontologias definidas, reutilizando e estendendo o Modelo de Proveniência Aberto (*Open Provenance Model* - OPM) [196], como forma de proporcionar interoperabilidade semântica, para geração de informações de proveniência quanto às séries temporais e aos métodos de *detrending* aplicados.
- Apresentar um estudo de caso para demonstrar, por meio de consultas na base de conhecimento, como ontologias permitem enriquecer semanticamente a análise de séries temporais, ao invés de fazer uso somente de metadados, contribuindo para a tomada de decisão quanto ao uso de métodos para sua correção.

1.3 Justificativa e Relevância

A justificativa para o desenvolvimento desta tese é a necessidade geração de conhecimento semântico quanto à aplicabilidade de métodos de *detrending* em séries temporais, objetivando facilitar a tomada de decisão em um processo de análise. A contribuição inédita desta pesquisa é o desenvolvimento modular de ontologias, centrado no reuso, permitindo interoperabilidade, reusabilidade, adaptabilidade e extensibilidade das ontologias.

Após pesquisas em bibliografias relacionadas, busca por ontologias existentes e, considerando a Tabela 7.2 do Capítulo 7 deste documento, não foram encontrados trabalhos com o objetivo de modelar e enriquecer semanticamente as características intrínsecas das séries temporais e processos de extração de tendências, caracterizando o ineditismo da tese.

1.4 Escopo da Pesquisa

O escopo desta pesquisa relaciona-se com a modelagem de métodos estatísticos que podem ser usados para extração de tendências em séries temporais no domínio do tempo. Destaca-se a possibilidade de extensão para descrição de métodos no domínio da frequência. Vale ressaltar que demais métodos da fase de pré-processamento das séries temporais que podem ser relacionados com *detrending*, tais como detecção e remoção de *outliers*, *clustering*, entre outros, não estão incluídos na ontologia em questão. A extração da sazonalidade, outro passo do pré-processamento, é tratado como um procedimento à parte, não sendo, portanto, abordado nesta pesquisa. Uma única aplicabilidade também abordada no modelo é o uso de filtros para remoção de ruído das séries temporais, dado seu relacionamento com métodos usados para *detrending* no referido contexto.

1.5 Estrutura da Tese

Além dessa introdução, os Capítulos 2 a 4 descrevem, respectivamente, a fundamentação teórica sobre séries temporais e métodos de *detrending*, ontologias e proveniência. O Capítulo 5 descreve os materiais e métodos utilizados. O Capítulo 6 apresenta a definição do modelo de proveniência para extração de tendências em séries temporais. O Capítulo 7 descreve o processo de avaliação das ontologias e apresenta uma análise comparativa dos trabalhos correlatos com o modelo definido. Por fim, o Capítulo 8 apresenta as conclusões e perspectivas de pesquisas futuras.

CAPÍTULO 2

SÉRIES TEMPORAIS

2.1 Introdução

Este capítulo apresenta duas seções principais. A primeira seção aborda a fundamentação teórica sobre séries temporais, incluindo definições e características, componentes e modelos de decomposição, análise e o problema de extração de tendências. A segunda seção descreve os métodos de estimação e extração de tendências, incluindo métodos paramétricos, relacionados com a tendência determinística e não-paramétricos, incluindo o uso de filtros digitais. Tendências estocásticas também são abordadas, assim como métodos de extração de ruído. Por fim, são feitas considerações quanto ao uso de métodos paramétricos ou não-paramétricos para o problema de extração de tendências.

2.2 Definições e Características das Séries Temporais

Matematicamente, Spiegel [230] define uma série temporal pelos valores Y_1, Y_2,... de uma variável Y nos tempos t_1, t_2,... Dessa forma, Y é uma função de t denotada por $Y = F(t)$. Uma série temporal envolvendo uma variável Y é representada por um gráfico de Y em função de t. A Figura 2.1 apresenta um exemplo de séries temporais onde a taxa mensal de inflação varia sobre o tempo.

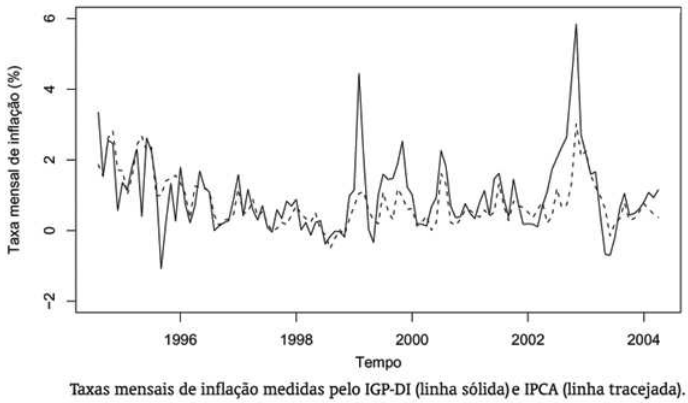


Figura 2.1: Exemplo de séries temporais [230].

Morettin e Toloi [198] definem uma série temporal como a realização de um processo estocástico. Seja T um conjunto arbitrário, um processo estocástico é uma família $Z(t)$, $t \in T$ tal que, para cada $t \in T$, $Z(t)$ é uma variável aleatória. Um processo estocástico é uma família de variáveis aleatórias, supostamente definidas num mesmo espaço de probabilidades. T é normalmente o conjunto dos números inteiros ou o conjunto dos números reais. Para cada $t \in T$, $Z(t)$ será uma variável aleatória real (Figura 2.2) [198].

A Figura 2.2 apresenta a interpretação de um processo estocástico, onde para $t \in T$, Z(T) é uma variável aleatória definida sobre o espaço de probabilidades, ou seja, $Z(t)$ é uma função de dois argumentos: $Z(t,w)$, $t \in T$, $w \in$ ao espaço de probabilidades. Para

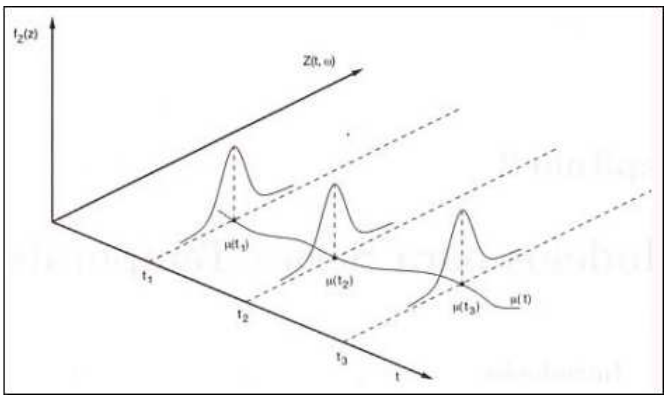


Figura 2.2: Processo estocástico como uma família de variáveis aleatórias [198].

cada $t \in$ T, tem-se uma variável aleatória Z(t,w) com uma distribuição de probabilidades, sendo possível que a função densidade de probabilidade (fdp) no instante *t1* seja diferente da fdp no instante *t2*, para dois instantes *t1* e *t2* quaisquer, mas geralmente a fdp de Z(t,w) é a mesma para todo $t \in$ T. Para cada w pertencente ao espaço de probabilidades, obtém-se uma função de t, uma realização ou trajetória do processo, ou seja, uma série temporal [198].

Segundo [97], séries temporais são dados observacionais gerados a todo instante, correspondendo a um volume sem precedentes, em muitas áreas do conhecimento. Constituem-se de uma sequência de pontos espalhados sobre o tempo, obtida geralmente em intervalos regulares [202, 207, 108].

A análise de séries temporais [223] promove a geração do conhecimento científico, pois podem ser extraídas informações a partir dos dados observados, em várias áreas da ciência. A metodologia estatística relacionada a dados observacionais é referida como Análise de Séries Temporais e, segundo Chandler e Scott [95] e Cryer [108], dados de séries temporais são caracterizados pela forma como foram gerados e coletados.

Existem dois tipos de dados de séries temporais [202], tempos contínuo e discreto. As séries temporais de tempo contínuo caracterizam observações realizadas a todo instante, como em instrumentos da área médica ou, na engenharia, quando sinais elétricos são registrados continuamente no tempo. Entretanto, séries temporais contínuas podem ser amostradas em pontos de tempo discretos, sendo tratadas como séries discretas. As séries de tempo discreto são caracterizadas por serem obtidas de forma usual em intervalos regularmente espaçados, como em um volume de vendas, onde dados são obtidos em intervalos específicos de tempo.

Nos dados de séries temporais reais, como em estudos ambientais, podem haver dados observacionais ausentes ou irregularmente espaçados, necessitando de análise da qualidade dos dados, devido a dificuldades de medição a partir de instrumentos científicos, havendo algum erro associado. A necessidade de modelar tais características é uma consideração importante para determinar uma estratégia adequada para a análise, juntamente com várias outras questões estatísticas [95]. Em séries temporais irregularmente espaçadas, há necessidade do uso de métodos especificamente desenvolvidos para esse caso ou, é necessária alguma transformação nos dados.

2.3 Componentes e Modelos de Decomposição das Séries Temporais

Séries temporais apresentam movimentos que são variações características, sendo que alguns ou todos podem estar presentes, em graus diversos. Esses movimentos característicos são comumente denominados componentes de uma série temporal. São quatro os principais componentes clássicos [183, 202, 230] (Figura 2.5): sazonais ou estacionais (S), cíclicos (C), irregulares ou aleatórios (I) e tendências (T).

Sazonal é o componente de variação em uma série temporal, é dependente de um período, descrevendo qualquer flutuação irregular com um período de menos de um ano. O componente cíclico corresponde a variações cíclicas de natureza não-sazonal, cuja periodicidade é desconhecida. O componente irregular são os resíduos aleatórios ou caóticos de ruído que sobram quando outros componentes da série, tais como tendência e sazonalidade, são removidos.

O componente tendência é o foco desta tese, sendo comumente definida como uma variação na média, na variância ou co-variância das séries temporais [97]. Tendências relacionam-se com a direção geral que a série temporal se desenvolve, ao longo do tempo, caracterizando um movimento secular [230]. Em ciências ambientais, a tendência é definida como uma variação de longo prazo nas propriedades estatísticas de um processo, onde o período da tendência é dependente de cada aplicação [95]. A Figura 2.3 apresenta tendências linear, quadrática, curva em formato S e exponencial.

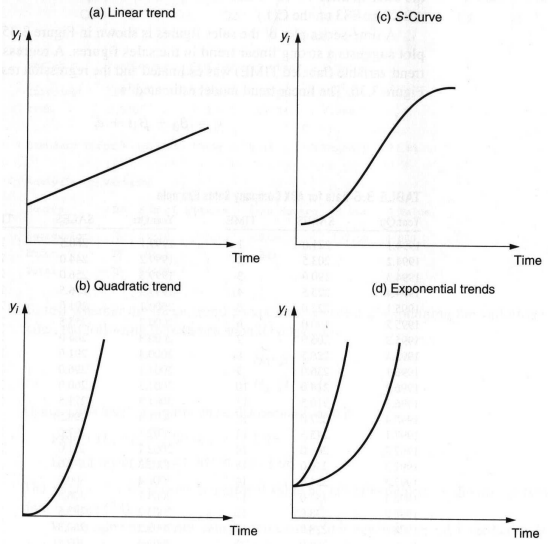


Figura 2.3: Tendências a) linear, b) quadrática, c) curva-S e d) exponencial [223].

A tendência necessita ser extraída das séries temporais não-estacionárias (Figura 2.4b) porque a mesma pode ocultar outros fenômenos, assim como a maioria dos métodos estatísticos é desenvolvida para séries temporais estacionárias (Figura 2.4a) [223], ou seja, que não apresentam tendências [193]. Tornar uma série temporal estacionária significa extrair todas as suas características determinísticas como as medidas estatísticas de média e variância, de forma que as correlações tornem-se independentes sobre o tempo [183].

A análise de séries temporais consiste em uma descrição matemática dos seus movimentos componentes. Uma das técnicas empregadas é considerar a variável Y como produto das Variáveis T, C, S e I: Y = T x C x S x I = TCSI. A Figura 2.5 apresenta os

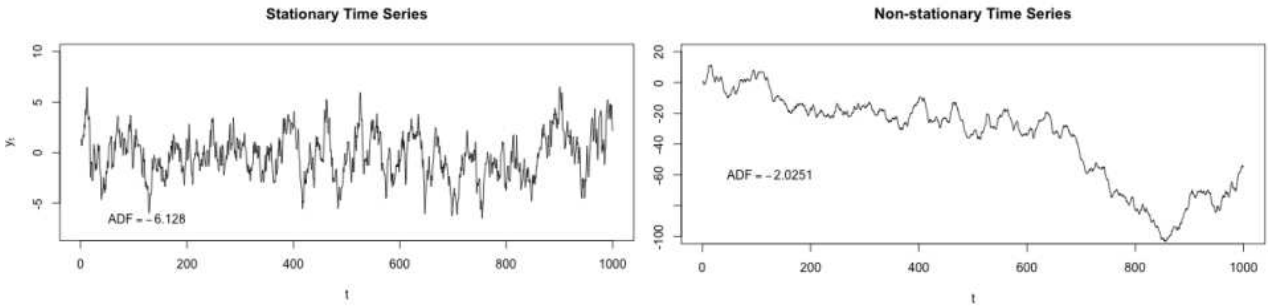


Figura 2.4: Séries temporais estacionárias e não-estacionárias [223].

componentes de uma série temporal. Em (a) tem-se um gráfico de uma reta de tendência a longo prazo ou secular; em (b) tem-se tendência a longo prazo ou secular e movimentos cíclicos (periódicos); e em (c) existem movimentos sazonais, tendência a longo prazo e movimentos aleatórios.

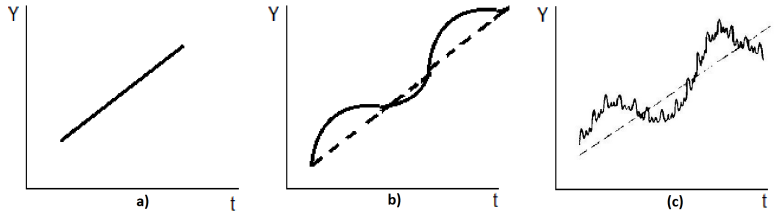


Figura 2.5: Componentes de uma série temporal [230].

Sobre os componentes das séries temporais, é relevante a identificação de informações, tais como [95]: tipo do componente, período, modelo e comportamento. Em relação ao tipo do componente tendência, a mesma pode ser determinística ou estocástica. No caso de uma série temporal apresentar tendência determinística, esta pode ser completamente determinada, uma vez que todas as quantidades relevantes, tais como os parâmetros e as variáveis explanatórias são conhecidos. Por outro lado, a tendência pode ser considerada estocástica, variando entre realizações de um processo. Quanto ao período, no caso da tendência, é relevante saber se a mesma é de curto ou longo prazo, podendo variar conforme a área. O modelo do componente, como no caso da tendência, este pode ser analisado parametricamente, fazendo uso de um modelo linear ou não-linear. O comportamento é relacionado se a tendência é monotônica, nesse caso somente evolui em uma dada direção; periódica, quando o padrão da tendência pode se repetir na série, entre outros.

Além dos componentes clássicos, em muitas séries temporais reais ocorrem eventos extremos, tais como *strikes*, *outliers*[1], saltos aleatórios (*jumps*), entre outros, sendo necessário avaliar o efeito desses eventos extremos, pois são características indesejáveis, as quais geralmente não são incluídas na análise [183].

A análise de séries temporais consiste em investigar os Fatores (T, C, S e I), sendo classificada como a decomposição de uma série temporal em seus movimentos componentes básicos. Ao analisar uma série temporal, estudam-se cada um desses componentes separadamente, excluindo o efeito dos demais.

[1]Em estatística, *outlier*, ou valor atípico, é uma observação que apresenta um grande afastamento das demais observações (que está "fora" dela), ou que é inconsistente. URL: http://dbpedia.org/page/Outlier

A decomposição de uma série temporal permite identificar importantes características dos dados. Modelos de decomposição clássicos são: multiplicativo, aditivo ou pseudo-aditivo [3]. Em séries que apresentam sazonalidade, o modelo aditivo é útil quando a variação sazonal é relativamente constante sobre o tempo, por outro lado, o modelo multiplicativo é útil quando a variação sazonal aumenta sobre o tempo [30].

Dentre as formas de decomposição das séries temporais, a decomposição contendo o componente tendência e os componentes sazonais e irregulares caracterizam a decomposição clássica, dividindo as séries em tais elementos [244]. Por outro lado, séries sazonalmente ajustadas são obtidas por estimar e remover efeitos sazonais a partir das séries temporais. Estimativas de séries sazonalmente ajustadas podem ser expressas como séries observadas menos o componente sazonal, ou seja, apresentam os componentes tendência e irregular.

2.4 Análise de Séries Temporais

A análise de séries temporais ocorre geralmente em duas fases, pré-processamento e análise de dados, onde ambas contém passos de processamento para obtenção do conhecimento científico. A análise de séries temporais é diferente da análise de dados tradicional, dada sua natureza intrínseca, onde observações são dependentes ou correlacionadas e a ordem das observações é importante na análise, ou seja, a análise de um dado no instante de tempo t é influenciada pela análise de dados em instantes anteriores.

Assim, procedimentos estatísticos tradicionais, baseados na suposição de dados independentes e identicamente distribuídos (*iid*), não se aplicam, sendo necessários diferentes métodos de análises [108]. Segundo Chandler e Scott [95], a escolha de métodos apropriados para análises depende das questões de interesse, onde o conhecimento sobre os dados das séries temporais é considerado essencial. Na análise de séries temporais, a dependência entre sucessivas observações é referida como auto-correlação e, segundo Chandler e Scott [95], é preferível a utilização de métodos de análise que são especificamente desenvolvidos para uso com dados auto-correlacionados.

Um primeiro passo em qualquer análise de séries temporais é a observação dos dados plotados sobre o tempo por pesquisadores. Este procedimento frequentemente sugere os métodos estatísticos de análise, assim como as estatísticas que sumarizam informações sobre os dados. Entre os objetivos para analisar séries temporais, destacam-se o entendimento e descrição do mecanismo gerador das séries, a previsão de valores futuros e o controle ótimo de um sistema [244]. A análise pode envolver séries temporais univariadas, onde uma única variável é medida sobre o tempo ou, multivariadas, onde mais de uma variável é medida simultaneamente [202].

A análise univariada está relacionada com várias suposições (hipóteses) para análise das séries [144], as quais são também utilizadas na análise multivariada. Tais suposições podem ser identificadas nas séries, a partir da análise visual, em gráficos das séries plotados em relação ao tempo. Para exemplificar, ao analisar um gráfico contendo observações feitas em intervalos regulares de tempo, rupturas em uma linha indicam observações ausentes. Como complemento à análise visual, testes de hipóteses estatísticos podem ser realizados para testar tais suposições. Outro exemplo são testes de hipóteses estatísticos para verificar se existem tendências nas séries temporais, os quais podem ser aplicados antes ou depois da estimação da tendência [198, 70, 95]. Na sequência são descritas algumas suposições sobre as séries temporais, as quais geralmente são verificadas em uma análise exploratória dos dados.

- Monotonicidade: uma série temporal é dita monotônica se consistentemente aumenta ou diminui [202].

- Sazonalidade: é uma característica de muitos conjuntos de dados de séries temporais. Um período sazonal pode ser definido como um padrão auto-repetindo [198].

- Homogeneidade: conceito relacionado à uniformidade ou sua ausência, heterogeneidade em um objeto. Um objeto que é homogêneo é uniforme em composição. Homogeneidade, segundo Meinl [183], significa que para uma dada série, todos os passos de tempo são igualmente espaçados. A maioria dos métodos estatísticos é desenvolvida para séries temporais regularmente espaçadas. Caso as séries sejam irregularmente espaçadas (não-homogêneas), é necessário um passo de pré-processamento usando métodos de interpolação a fim de regularizar os dados originais. Embora existam métodos para tratar especificamente esse tipo de observação, destaca-se que a maioria dos modelos considera séries homogêneas. Um exemplo é o método de médias móveis ponderadas, que não é compatível com dados irregularmente espaçados.

- Homoscedasticidade: uma sequência ou um vetor de variáveis randômicas é homoscedástico (variância constante) se todas as variáveis randômicas têm a mesma variância finita (homogeneidade da variância). O complemento é heteroscedasticidade (variância não-constante)[2].

- Normalidade: dados normais são dados obtidos a partir de uma população que tem uma distribuição normal, a qual é a mais comumente usada na teoria e na prática da estatística [203].

- Estacionariedade: em uma série temporal estacionária, a função média deve ser constante no tempo [108]. Tornar uma série temporal estacionária está entre as primeiras e mais importantes tarefas na análise de séries temporais [183].

Séries temporais podem ser analisadas no domínio de tempo ou de frequência (Figura 2.6). Nesse último, a análise é conhecida como Análise de *Fourier*, onde o componente é analisado sob o sinal inteiro e informações sobre o domínio do tempo são perdidas. Quando esse domínio é obrigatório, métodos *wavelets* têm se destacado, especificamente desenvolvidos para análises tempo-frequência, considerando as duas dimensões ao mesmo tempo, decompondo o sinal em diferentes escalas [183].

Na análise de séries temporais, é relevante conhecer o processo gerador dos dados, podendo ser estocástico ou determinístico. Processos estocásticos denotam um processo onde o movimento a partir de um estado para o outro é determinado por uma variável independente nos estados inicial e final [13]. Se o processo é estocástico, cada valor de dado das séries pode ser visualizado como uma simples média de uma distribuição de probabilidade de uma população em cada ponto de tempo. Cada distribuição tem uma média e uma variância. Por outro lado, séries temporais que não são descritas por processos estocásticos são geradas por processos determinísticos, onde alguns processos determinísticos podem ter relacionamentos funcionais. Pode haver qualquer número de processos que não envolvem distribuição de probabilidade e estimação [249].

Processos estocásticos podem ser estacionários ou não-estacionários. A estacionariedade é uma propriedade estatística a ser considerada na análise de séries temporais, pois

[2]DBpedia. URL:http://dbpedia.org/page/Homoscedasticity

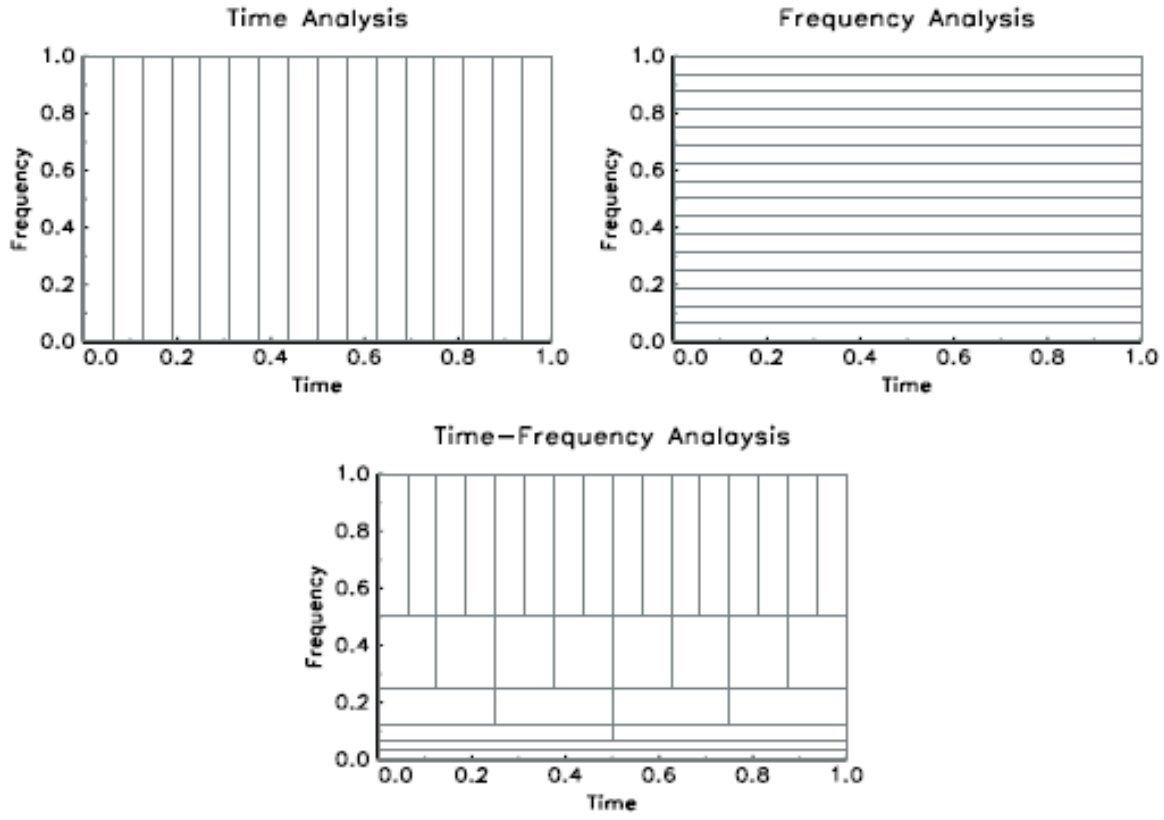


Figura 2.6: Domínio de análise das séries temporais [72].

a maioria dos modelos estatísticos requer que os processos sejam estacionários. Se um processo é não-estacionário, uma transformação nos dados pode ser feita para se conseguir estacionariedade. A maioria dos dados reais são não-estacionários, ou seja, apresentam algum tipo de tendência que deve ser removida [107]. Segundo [70], se tendências não forem removidas, grandes distorções podem ocorrer em processamentos posteriores da densidade de probabilidade, correlação e quantidades espectrais.

Séries temporais não-estacionárias com mudanças na média ou variância/co-variância são comuns em muitas áreas do conhecimento. Nesse caso, a estacionariedade (estabilidade) das séries é requerida, implicando em alguma forma de remoção da não-estacionariedade. A escolha de um procedimento de *detrending* correto depende da causa da não-estacionariedade [232].

Um processo é dito ser estacionário se sua média, variância e co-variância não mudam sobre o tempo, caso contrário, o processo é não-estacionário, onde ambas as medidas podem estar presentes, havendo a necessidade de transformação dos dados para se atingir estabilidade. O método de transformação depende da causa da não-estacionariedade. Segundo Stadnytska [232], existem dois métodos para estabilizar séries temporais: *differencing* e regressão mínimos quadrados ordinários.

A Figura 2.7 mostra exemplos de processos não-estacionários e suas funções de Auto-Correlação (ACF) e Auto-Correlação Parcial (PACF). Nos três exemplos, o termo erro da equação é independente e identicamente distribuído com média zero e variância constante ($\mu \sim (0,\ \sigma^2)$). Em (a), o processo é chamado um passeio aleatório puro (*pure random walk*), onde a média é igual, mas a variância ($t\sigma^e$) aumenta indefinidamente sobre o tempo, o qual pode ser representado como a soma de choques aleatórios (*random shocks*).

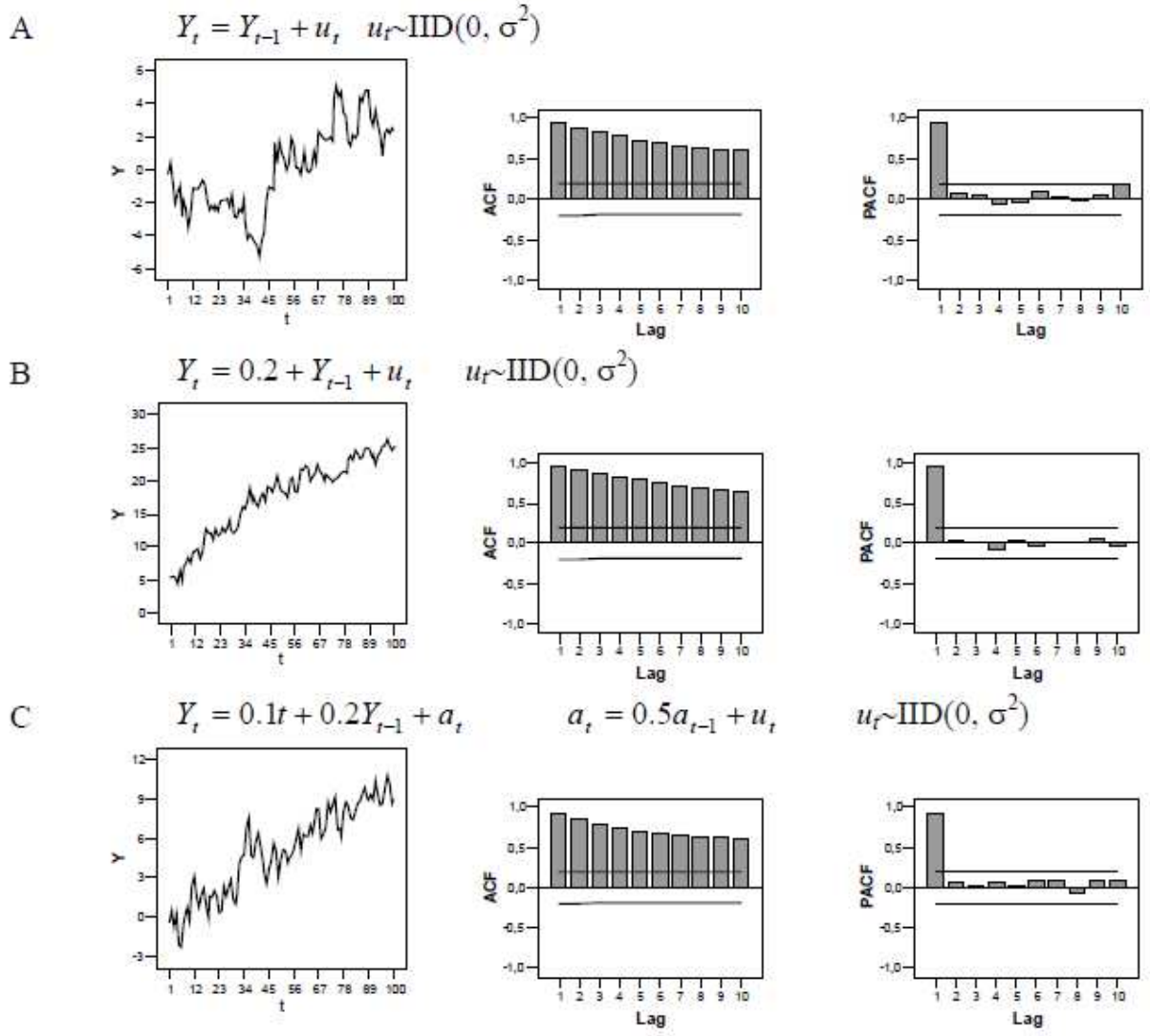


Figura 2.7: Processos não-estacionários. ACF e PACF. (a) Passeio aleatório puro, (b) passeio aleatório com desvio e (c) tendência de tempo determinística [232].

Em (b), um termo constante é adicionado na referida equação Y_t, chamado passeio aleatório com desvio (*random walk with drift*), onde o termo constante é o parâmetro do desvio e, dependendo se é positivo ou negativo, é exibido uma tendência estocástica negativa ou positiva. Nesse caso, tanto a média quanto a variância aumentam sobre o tempo. Nos processos de passeios aleatórios, estes são não-estacionários, onde a primeira diferença (*first differencing*) é estacionária. Ambos os tipos de passeios aleatórios (com ou sem desvio) são processos estacionários de diferença (*Difference Stationary Process* - DS). Modelos desse tipo são conhecidos como processos de raiz unitária[3], sendo um caso específico de uma classe mais geral de modelos estocásticos conhecidos como processos integrados. Se a ordem for 1, I(1) são ditos integrados de ordem 1 na metodologia de Box e Jenkins (70) [79]. Se uma série temporal necessita ser diferenciada d vezes para se tornar estacionária, a série é dita integrada de ordem d.

Em (c), a equação é determinada por uma tendência de tempo determinístico, onde o

[3]o termo raiz unitária refere-se à raiz do polinômio no operador *lag* (L), onde $\text{LY}_t = Y_{t-1}, L^2Y_t = Y_{t-2}$ e, se (1-L) = 0, obtém-se L = 1, por isso a denominação raiz unitária [232].

processo apresenta uma variância estável e uma mudança na média. A média é a parte ($\beta_1 + \beta_2 t$) da equação. Ao subtrair a média da série original, as séries resultantes são estacionárias, processo conhecido como *detrending* polinomial, gerado por um processo estacionário de tendência (*Trend Stationary Process* - TS).

O método de transformação apropriado quanto à não-estacionariedade depende do processo gerador dos dados [232]. No caso de uma dada série temporal ser considerada como a realização de um processo de passeio aleatório, a solução para estacioná-la é diferenciá-la uma vez (*first differencing*). Caso contrário, se a série é estacionária ao redor de uma linha de tendência, a maneira correta de transformá-la é regredi-la no tempo, onde os resíduos gerados a partir da regressão serão estacionários [232]. Na Figura 2.7, é difícil decidir se a tendência é determinística ou estocástica, pois as realizações dos processos DS e TS parecem muito similares. Nesse caso, Stadnytska [232] destaca que transformações inapropriadas são comuns na prática, explicando quais as consequências desse fato, como o caso de fazer uma regressão em um passeio aleatório, onde resíduos não são estacionários e tendem a exibir ciclos, nesse caso, um método de *detrending* aplicado no tipo de tendência incorreto é chamado *underdifferencing* e no caso de resíduos inapropriadamente gerados por TS é chamado *overdifferencing*, o qual é considerado um erro menos sério do que no primeiro caso.

Uma forma de representar tendências é quando a mesma é considerada fixa e observada imperfeitamente, devido a um componente ruído. Entretanto, esta não é a única forma de representar tendências. Quando a variação é controlada unicamente através da dependência entre sucessivas observações, ao invés de uma tendência fixa, onde outra simulação do mesmo modelo mostra características similares, mas com os tempos de picos e quebras na estrutura diferentes, essa tendência aparente pode ser representada como Estocástica [84], variando entre realizações de um processo. Os processos não-estacionários geradores de tendência estocástica são denominados processos de Diferença-Estacionária (*Stationary Difference*), onde a mesma é geralmente removida pelo método da Diferença.

Determinar se uma tendência é do tipo determinístico ou estocástico não é uma tarefa trivial, onde o conhecimento do sistema físico pelo pesquisador influencia nas decisões. Uma tendência determinística é completamente previsível e não variável, onde *detrending* polinomial pode ser um meio correto de conseguir estacionariedade. Uma tendência estocástica não é previsível, a qual necessita ser diferenciada para se tornar estacionária. Stadnytska (2010) também destaca que é difícil distinguir entre os tipos de tendências visualmente ou pela análise das funções ACF e PACF das séries temporais, citando a necessidade de aplicação de Testes de Raiz Unitária como o *Augmented Dickey-Fuller*, onde, se não existe uma raiz unitária, significa que a série é integrada I(0) ao redor de uma tendência determinística. Caso contrário, se a série apresentar uma raiz unitária, o processo é integrado I(1) e a tendência é considerada estocástica [84]. A simultânea existência de uma raiz unitária e uma tendência determinística é considerada como algo não-realístico em [232].

Existe uma variedade de modelos para representar tendências estocásticas. Estes podem ser usados para representar a estrutura em séries temporais ou, serem combinados, para representar estrutura que não é descrita por tendências determinísticas [95]. Essa modelagem não é o foco desta tese, mas modelos considerando tendências estocásticas podem ser estendidos na ontologia. A próxima seção descreve o problema de extração de tendência em séries temporais, incluindo a descrição dos métodos que podem ser usados para realizar esse objetivo.

2.5 Problema de Extração de Tendências em Séries Temporais

Na literatura de séries temporais existem muitas definições para o componente tendência, destacando-se como definição usual uma variação no nível médio das séries temporais. Esta seção aborda várias outras definições, dependentes do contexto de análise. O conhecimento sobre a definição do conceito tendência tem implicações no método a ser utilizado para sua extração, entre outras questões, como o conhecimento do sistema físico.

Chatfield [97] descreve tendência como uma mudança de longo prazo no nível médio das séries temporais. Em ciências ambientais, Chandler e Scott [95] definem a tendência como uma variação temporal de longo prazo nas propriedades estatísticas de um processo, onde o prazo depende da aplicação. A escolha de um método apropriado de análise depende das questões de interesse. Em aplicações ambientais, possíveis razões para a análise de tendências são os sistemas onde mudanças em longo prazo podem obscurecer aspectos de interesse real: i) Descrever o comportamento passado de um processo; ii. Tentar entender os mecanismos por trás de mudanças observadas; iii. Fazer avaliações de possíveis cenários futuros, extrapolando as alterações passadas; iv. Monitorar a eficácia das políticas de controle ambiental; e v. Permitir a análise de sistemas, onde as mudanças de longo prazo servem para obscurecer aspectos de interesse real. Nessa última questão, um primeiro passo na análise é a identificação e remoção de tendências para ver mais claramente os inter-relacionamentos dos dados. Esta tese está relacionada com essa questão, onde a tendência é claramente definida e necessita ser extraída, caso contrário, é mais difícil a descoberta do conhecimento científico, nas diversas áreas que analisam séries temporais.

Encontrar, identificar, modelar e remover tendências correspondem às tarefas de análise e predição de séries temporais. Essas tarefas têm sido aplicadas em diversas áreas do conhecimento. Montesino-Pouzouls e Lendasse [193] afirmam que encontrar tendências pode ser motivado por duas principais razões: i. Transformar uma série temporal não-estacionária para uma estacionária; e ii. Caracterizar seu comportamento pela separação de componentes, como a própria tendência, os ciclos, as flutuações e os ruídos. Segundo os autores, não há um consenso geral de como tendências deveriam ser modeladas. É possível afirmar que não há uma definição universal de tendência relacionada a todos os campos de aplicação [248, 193]. É aceito que uma tendência é um componente que evolui lentamente em longo prazo sob o sistema.

Em linguagem de frequências, Meinl [183] caracteriza tendência como sendo limitada, para certas baixas frequências, excluindo quaisquer influências de ruídos e flutuações, a partir de mais altas frequências. Essa definição não é satisfatória para muitas séries temporais. Modelos teóricos não incorporam aspectos como sazonalidade ou até mesmo saltos (*jumps*), sendo ainda amplamente usados porque assumem uma divisão perfeita entre tendências e flutuações estocásticas. Ao considerar tendências sob períodos mais longos, aparecem significantes *jumps* ou grandes declives (*steep slopes*) que não podem ser atribuídos como parte de ruído estocástico persistente, causados por fatores externos.

O restante desta seção é fundamentada em [248], onde os autores citam que, apesar de amplamente utilizado, o conceito de tendência não é formalmente definido. Segundo os autores, o problema de extração de tendências, a partir de séries temporais, impõe muitas questões. Geralmente, métodos para análise e predição de séries temporais podem ser afetados por passos de pré-processamento, faltando um consenso sobre como esses métodos deveriam ser aplicados, ou se tendências deveriam ser modeladas, separadamente, dos demais componentes. A tendência geralmente é definida envolvendo parâmetros ou funções de modos extrínsecos e pré-determinados, com formas funcionais pré-seleciona-

das, onde uma simples linha de tendência pode ser ajustada aos dados. Nesse caso, a extração da tendência consiste em remover uma linha reta de melhor ajuste aos dados, produzindo resíduo de média zero. Outra abordagem para tendência comumente usada é a média em movimento, a qual requer uma escala de tempo pré-determinada para efetuar a operação da média. Métodos mais complexos de *detrending*, tais como análise de regressão ou filtros baseados em *Fourier* são frequentemente baseados em suposições lineares, tornando difícil justificar seu uso em dados não-lineares e não-estacionários. Mesmo no caso de quando a tendência é calculada, a partir de uma regressão não-linear, é difícil justificar a escolha de uma fórmula de regressão, independente do tempo, para aplicá-la globalmente a processos não-estacionários. Em geral, vários ajustes de curvas com fórmulas funcionais determinadas *à priori* são subjetivas, exceto para os casos em que processos físicos são completamente conhecidos. Segundo esses autores, os métodos de regressão, médias móveis e filtros são todos problemáticos ao modelar dados não-lineares e não-estacionários. Nesse cenário se destaca o Método Modo de Decomposição Empírica (*Empirical Mode Decomposition* - EMD) [156].

A maioria dos métodos disponíveis envolvem parâmetros ou funções com formas funcionais pré-selecionadas, sendo extrínsecas e subjetivas. Essa definição de tendência é puramente linear e estacionária, podendo ser ilógica e fisicamente sem sentido em aplicaçõ-es do mundo real, onde o ajuste linear faz pouco sentido para este caso, sendo não-linear e não-estacionário. Nesse caso, duas questões são observadas, conforme [248]: i. A tendência deveria ser uma propriedade intrínseca dos dados, sendo uma parte integral dos mesmos. O método usado para sua definição precisa ser adaptativo, onde a tendência extraída é derivada dos dados; ii. A tendência deveria existir dentro de um dado intervalo de dados (intervalo inteiro ou uma parte) e ser uma propriedade associada com a correspondente escala de tempo local. A adição contínua de novos dados tem efeitos, levando a uma extensão da tendência atual, com uma escala de tempo maior, onde os dados atuais se inserem. Ou seja, nesse caso a tendência é uma das propriedades locais dos dados e, portanto, tem de ser associada com uma escala de tempo para não ser confundida com ciclos locais.

Tendência é definida de forma intrínseca e adaptativa, em processos não-estacionários e não-lineares sem depender de funcionalidades extrínsecas. Sendo intrínseca, requer que o método usado para definir a tendência seja adaptativo para ajustar dados a partir de processos não-estacionários e não-lineares. A definição de tendência para não-estacionariedade e não-linearidade é dada por Wu et al [248] como uma função monotônica ajustada intrinsecamente ou uma função em que pode haver no máximo um extremo dentro de um dado intervalo de dados. Um intervalo de dados pode ser o intervalo todo, ou uma parte dos dados. *Detrending* é o processo definido como a operação de remover a tendência e variabilidade é definida como o resíduo dos dados, após a remoção da tendência, dentro de um dado intervalo de dados.

Apesar da importância da extração de tendências em séries temporais, a literatura não contempla muitos resultados teóricos que sejam suficientemente genéricos, sendo na maioria dos casos específicos da aplicação e/ou baseados em dados limitados ou restritos para uma determinada abordagem. Wu et al [248] relatam que os resultados disponíveis são dispersos, mostrando casos *ad-hoc*, baseados em dados sintéticos ou limitados do mundo real. Na abordagem mais simples, a tendência é identificada por ajustar um componente determinístico, geralmente linear. Nesse caso, a tendência é subtraída da série temporal para garantia de estacionariedade. Uma característica deste processo é que a série resultante deve ser um processo de média zero para o período de tempo.

2.6 Métodos de Estimação e Remoção de Tendências em Séries Temporais

Esta seção descreve os métodos de estimação e remoção de tendências, incluindo métodos paramétricos, relacionados com a tendência determinística e não-paramétricos, incluindo o uso de filtros digitais. Métodos de remoção de tendências são abordados, incluindo a descrição de dois métodos no domínio tempo-frequência, os quais são utilizados como exemplo da extensibilidade da Ontologia DO. Destaca-se que tais métodos podem ser usados para *detrending* de forma robusta, ou seja, são suscetíveis à ocorrência de *outliers* nos dados [203].

2.6.1 Métodos de Estimação de Tendência Paramétricos

São apresentados na sequência os métodos de estimação de tendências de forma paramétrica, incluindo métodos de análise bivariada simples (regressão linear e não-linear) e análise multivariada (regressão múltipla). Nesse caso, a tendência é considerada determinística, podendo ser completamente ajustada e removida dos dados.

2.6.1.1 Análise de Regressão Bivariada

A análise de regressão bivariada mede a associação entre uma variável resposta Y e um conjunto de variáveis independentes $(X_1, X_2, ...X_n)$, estimando parâmetros do comportamento entre elas. A correlação determina o inter-relacionamento e dependências entre diferentes séries temporais e a auto-correlação determina o relacionamento dos dados da própria série. Esta análise fornece um número, denominado Coeficiente de Correlação[4], indicando como as variáveis variam conjuntamente, medindo a intensidade e a direção das relações (lineares ou não-lineares) entre as variáveis [230].

Baseado nos dados amostrais, o valor de uma variável Y pode ser estimado em relação ao valor conhecido de uma variável X, por meio da avaliação do valor de Y, a partir de uma curva de mínimos quadrados, ajustado aos dados amostrais. A curva resultante é denominada Regressão de Y para X. Caso a variável independente X corresponda ao tempo, os dados representam valores de Y em diferentes momentos. Esses dados ordenados em relação ao tempo são as séries temporais. A reta ou a curva de regressão de Y para X é denominada de Tendência [230].

- Análise de Regressão Linear

  O Modelo de Regressão Linear [91, 115] é um modelo matemático relacionando o comportamento de uma variável Y com outra X. Nesse modelo, a função f relaciona duas variáveis na forma f (X) = a + b X, onde a variável X é a variável independente da equação e Y = f (X) caracteriza a variável dependente das variações de X.

  Quando envolve uma relação causal entre duas variáveis, o modelo é denominado simples e, multivariado envolvendo uma relação causal com mais de duas variáveis, onde o comportamento de Y é explicado por mais de uma variável independente X_1, X_2, ...X_n.

  Independente de ser simples ou multivariados, os modelos simulam relacionamentos entre as variáveis, os quais podem ser lineares, com uma equação da reta ou do

[4] denotado pela letra (r) que advém de regressão

plano ou, não lineares com uma equação exponencial, geométrica, entre outras. Dessa forma, basicamente, a análise de regressão compreende os seguintes tipos de modelo: linear (simples ou multivariado) e não-linear (simples ou não-linear multivariado).

O Modelo de Regressão Linear Simples é caracterizado como o método mais comum para identificação de tendência, onde uma linha simples é ajustada aos dados. Tais modelos empiricamente desvendam relacionamentos complicados entre variáveis. Auxilia nas explicações de observações de uma variável dependente, denotada por y, com valores observados de uma ou mais variáveis independentes, denotadas por x_1, x_2, ... O termo Linear refere-se ao tipo de equação usada no modelo [179]. Em uma regressão simples, segundo [115], existem dois coeficientes de regressão, β_0 e β_1, existindo (n-2) graus de liberdade (*degress of freedom*), sendo n o tamanho da amostra. Se a relação entre duas variáveis é aproximadamente linear, dados podem ser ajustados por uma reta passando pelos pontos. A equação desta reta é dada pela Equação 2.1 [91]:

$$E(Y \mid x) = \alpha + \beta x \tag{2.1}$$

onde E(Y | x) é a esperança condicional de Y, uma função de x, ou seja, igual a μ(x), o intercepto α (representa o ponto onde a reta corta o eixo das ordenadas) e a inclinação β (coeficiente angular representando o quanto varia a média de Y para o aumento de uma unidade da variável X) são os parâmetros desconhecidos a serem estimados (Figura 2.8).

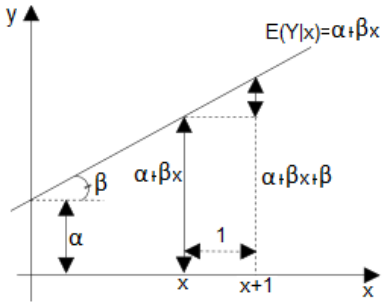


Figura 2.8: Representação do modelo [91].

Uma característica comum a todos os modelos de regressão é o termo Erro, sendo incluído para capturar fontes que não são obtidas por outras variáveis [179]. Determinam como duas variáveis se relacionam, estimam a função que determina a relação entre as variáveis e usam a equação de ajuste para prever valores da variável dependente. O modelo da Equação 2.1 é também composto por uma parte não-determinística, devido a fatores não observáveis, sendo um componente aleatório (variável aleatória ε). O modelo completo é dado pela Equação 2.2 [91]:

$$E(Y \mid x) = \alpha + \beta X + \varepsilon \tag{2.2}$$

onde ε tem as seguintes suposições assumidas para a variável aleatória: E(ε) = 0; Var(ε) = σ^2; $\varepsilon \backsim$ N(0, σ^2), onde E(Y | x) $\backsim$ N($\alpha + \beta$ x, σ^2).

Tais modelos são apropriados sob certas suposições: i. A relação entre X e Y é linear; ii. Valores de X são fixos, ou seja, X não é uma variável aleatória; iii. a Média dos erros é nula, ou seja, E(μ) = 0, i=1, 2,...n. iii. para um dado valor x de

X, a variância dos erros μ_i é sempre σ^2, denotada variância residual. Nesse caso, o erro é homocedástico; iv. O erro em uma observação é não-correlacionado com o erro em qualquer outra observação; e v. os erros apresentam Distribuição Normal.

Em casos mais gerais, o termo linear refere-se ao modo como os parâmetros entram no modelo, sendo de forma linear [91], como na Equação 2.3 que representa uma

$$E(Y \mid x) = \alpha + \beta x + \gamma x^2 \tag{2.3}$$

parábola, sendo linear nos parâmetros α, β e γ. O modelo da Equação 2.4

$$E(Y \mid x) = \alpha e^{\beta x} \tag{2.4}$$

não constitui um modelo linear em α e β. Vários modelos não-lineares surgiram para casos onde o modelo linear normal não é adequado. Modelos não-lineares podem ser transformados em lineares, por meio de transformação de variáveis. Tomando-se o logaritmo de base e, obtém-se (Equação 2.5),

$$lnE(Y \mid x) = ln(\alpha) + \beta x = \alpha' + \beta x \tag{2.5}$$

sendo linear em α' e β.

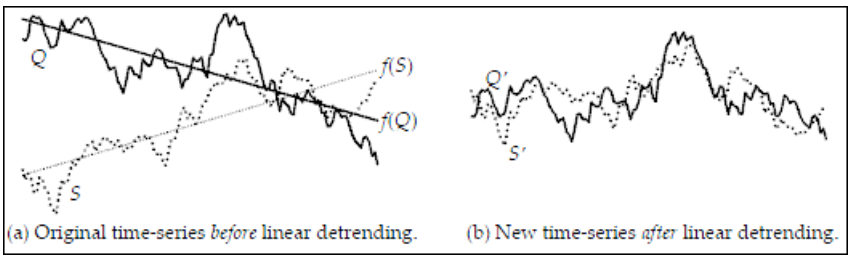


Figura 2.9: *Detrending* linear [91].

O processo onde essa linha reta é subtraída, a partir das séries temporais, produz um resíduo de média zero. É um caso particular de *detrending* polinomial, de ordem 1, usando função linear (Figura 2.9). A estimação dos parâmetros usa o método dos mínimos quadrados, no domínio do tempo. *Detrending* com polinômios de ordem superior podem apresentar problemas de super ajuste - *overfitting*.

Modelos probabilísticos buscam medir a variabilidade de fenômenos casuais, conforme suas ocorrências. Na prática, o pesquisador tem alguma ideia sobre a forma de distribuição, mas não dos valores exatos dos parâmetros que a especifica. É possível ter ideia de que a distribuição seja Normal, mas essa informação não é suficiente para determinar qual a distribuição Normal correspondente. É necessário conhecer os parâmetros, tais como a média e variância, para que seja completamente especificada. Dessa forma, o propósito do pesquisador é descobrir, ou estimar, os parâmetros da distribuição para sua posterior utilização. A solução é selecionar parte dos elementos (amostra), analisá-la e inferir propriedades para o todo (população) [91].

A obtenção de amostras de séries de dados temporais é feita por levantamento observacional, onde dados são coletados sem que o pesquisador tenha controle sobre

as informações obtidas, exceto quando ocorrem erros grosseiros. Nesse caso, a especificação de um modelo desempenha papel crucial na ligação entre dados e população. No caso de uma série temporal, o modelo subjacente é o de processo estocástico, onde a série observada é uma das infinitas realizações desse processo. A população hipotética é o conjunto de todas essas realizações e a série observada é a amostra [91].

- Análise de Regressão Não-Linear

  Em modelos de regressão linear e múltipla, as tendências nos dados deveriam ser explicadas pela dependência em co-variáveis ou consideradas como funções polinomiais do índice de tempo *t*. Embora sejam modelos lineares, tendências podem não ser lineares. Assim, existem tipos de tendências que não podem ser representadas dessa forma. Em um modelo não-linear, os parâmetros entram não-linearmente na função tendência. Para esse modelo, a tendência é não-linear [95].

  Modelos não-lineares podem ser ajustados usando mínimos quadrados da mesma forma que modelos lineares. A teoria relacionada é a mesma que para modelos lineares, entretanto, os resultados são baseados em aproximação de grandes amostras. A principal dificuldade é que não existe nenhuma solução explícita para estimativas de parâmetros, pois minimizar a soma dos quadrados deve ser feito numericamente, mas pode ser difícil computacionalmente, apesar de haver uma variedade de algoritmos disponíveis [95].

  Em ciências ambientais, Chandler e Scott [95] citam que modelos paramétricos explícitos, para tendências não-lineares, tendem a ser aplicáveis em domínios mais especializados, tais como o crescimento da população. Todavia, é difícil justificar o uso de uma representação paramétrica explícita para tendências não-lineares, pois pode ser mais adequado tratar esse tipo de relacionamento usando métodos não-paramétricos.

Quanto ao ajuste de funções, quando se tem duas ou mais variáveis, observa-se que existe uma relação entre elas. Muitas vezes essa relação é expressa em forma matemática, por meio de uma equação que relaciona as variáveis [230]. Para se determinar essa equação, são colecionados dados indicativos de valores correspondentes as variáveis. Considere X e Y representando, respectivamente, a altura e o peso de adultos do sexo feminino, uma amostra de N indivíduos apresenta as alturas X_1, X_2,...X_n e os pesos correspondentes Y_1, Y_2,...Y_n. O próximo passo é alocar esses pontos X e Y em um sistema de coordenadas cartesianas. O conjunto de pontos resultantes é denominado *diagrama de dispersão*.

No diagrama de dispersão, é possível observar uma curva regular (de ajuste), a qual se aproxima dos dados. Conforme a Figura 2.10, quando os dados parecem estar bem próximos de uma linha reta, diz-se que há uma relação *linear* entre as variáveis e *não-linear*, caso contrário.

Na sequência, estão relacionados tipos comuns de ajustes de funções e suas respectivas equações. A variável X é dita variável independente e a variável Y é dita variável dependente.[5] Todos os demais elementos das equações são constantes [230]. As Equações (2.6 a 2.10) são denominadas Polinômio do Primeiro, Segundo, Terceiro, Quarto e Enésimo Graus, respectivamente. As quatro primeiras funções são denominadas Funções Linear,

[5]Os papéis das variáveis independente e dependente podem ser trocados.

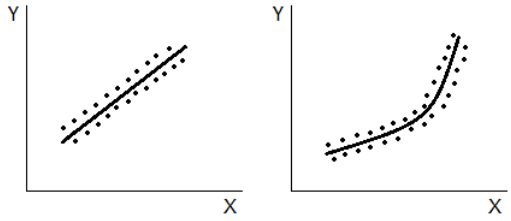


Figura 2.10: Regressão linear e não-linear entre as variáveis X e Y [230].

Quadrática, Cúbica e do 4° grau, respectivamente.

Linha reta:

$$Y = a_0 + a_1 X \tag{2.6}$$

Parábola ou Curva do 2° grau:

$$Y = a_0 + a_1 X + a_2 X^2 \tag{2.7}$$

Função do 3° grau:

$$Y = a_0 + a_1 X + a_2 X^2 + a_3 X^3 \tag{2.8}$$

Função do 4° grau:

$$Y = a_0 + a_1 X + a_2 X^2 + a_3 X^3 + a_4 X^4 \tag{2.9}$$

Função do enésimo grau:

$$Y = a_0 + a_1 X + a_2 X^2 + ... + a_n X^n \tag{2.10}$$

Muitas outras equações são possíveis de utilização, dentre as quais destacam-se:

Hipérbole:

$$Y = \frac{1}{a_0 + a_1 X} ou \frac{1}{Y} = a_0 + a_1 X \tag{2.11}$$

Função Exponencial:

$$Y = ab^X ou \log Y = \log a + (\log b)X = a_0 + a_1 X \tag{2.12}$$

Função Geométrica:

$$Y = aX^b ou \log Y = \log a + b \log X \tag{2.13}$$

Função Exponencial modificada:

$$Y = ab^X + g \tag{2.14}$$

Função Geométrica modificada:

$$Y = aX^b + g \tag{2.15}$$

Função de Gompertz:

$$Y = pq^{b^X} ou \log Y = \log p + b^X \log q = ab^X + g \tag{2.16}$$

Função de Gompertz modificada:

$$Y = pq^{b^X} + h \tag{2.17}$$

Função Logística:

$$Y = \frac{1}{ab^X} + gou\frac{1}{Y} = ab^X + g \tag{2.18}$$

Para decidir qual função utilizar, são utilizados diagramas de dispersão das variáveis transformadas. Para facilitar esse processo, pode-se utilizar uma escala *semilog* ou ambas as escalas logarítmicas *log-log* [230].

Em relação à Estimação de Parâmetros, a partir da obtenção de uma amostra, a mesma pode ser usada para produzir alguma característica específica. Uma Estatística T é uma característica da amostra [91], sendo uma estatística T é uma função de X_1, X_2, ..., X_n. Ou seja, são funções de valores amostrais, tais como a média, a variância, o menor valor, o maior valor, a amplitude amostral, entre outros.

Um parâmetro é uma medida usada para descrever uma característica da população [91], constituindo funções de valores populacionais. Ao se coletar amostras de uma população, identificada pela variável aleatória X, são parâmetros a média E(X) e sua variância Var(X). Jackman [159] ilustra os parâmetros da população e estatísticas da amostra (Tabela 2.1):

Tabela 2.1: Parâmetros da população e estatísticas da amostra [159].

	Parâmetro da População	Estatística (romano)
proporção	π	p
média	μ	$\bar{X}$
variância	σ^2	S^2
correlação	ρ	r
coeficiente de regressão	β	b

O problema da Inferência Estatística é fazer uma afirmação sobre os parâmetros da população através da amostra. Colhida a amostra, tem-se observado um particular valor de T, por exemplo t_0, e baseado nesse valor é feita a afirmação sobre o parâmetro populacional [91]. A distribuição amostral de T é dada por uma população X, com determinado parâmetro de interesse, onde todas as amostras são retiradas da população, conforme determinado procedimento e, para cada amostra, é calculado o valor t da estatística T e os valores t formam uma nova população, cuja distribuição recebe o nome de distribuição amostral de T.

A Inferência Estatística objetiva fazer generalizações sobre uma população, baseada nos dados de uma amostra. Dois problemas básicos são [91]: estimação de parâmetros e testes de hipóteses sobre parâmetros [115].

Um estimador é não-tendencioso quando a média da distribuição amostral de uma estatística for igual ao parâmetro populacional correspondente, caso contrário será tendencioso. Os valores correspondentes dessas estimativas são não-tendenciosas ou tendenciosas, respectivamente [230].

Se as distribuições amostrais de duas estatísticas têm a mesma média, a estatística de menor variância é o estimador eficiente da média, e as demais são denominados

estimadores ineficientes. Os valores correspondentes das estatísticas são estimativas eficientes e ineficientes, respectivamente [230]. As estimativas ineficientes são usadas devido a facilidade de obtenção das mesmas.

As estimativas podem ser [230]: por pontos, dada por um número único; por intervalos, indicando sua precisão, dada por dois números; e intervalos de confiança, baseados na distribuição amostral do estimador pontual, onde números extremos dos intervalos são os limites de confiança ou fiduciais. Os intervalos de confiança podem estar relacionados as médias, proporções, diferenças, somas e desvios padrões.

Conhecer as propriedades ou critérios dos estimadores é um dos propósitos da Inferência Estatística [91], pois não existe um único critério para a escolha de estimadores, mas sim um conjunto de critérios que podem ser usado para seleção e comparação [120]. É possível ter mais de um estimador para um mesmo parâmetro, onde é necessário saber qual deles é melhor. O julgamento pode ser feito pela análise de suas propriedades. As propriedades incluem *vies* (estimador não-viesado ou não-viciado) e consistência. Um estimador é não-viesado quando seu valor esperado coincide com o parâmetro de interesse. Um estimador é consistente quando seu valor se aproxima do verdadeiro valor do parâmetro, a medida que aumenta o tamanho da amostra [91].

A Inferência Estatística se baseia na teoria denominada Frequentista ou Clássica[6]. Um dos problemas em Inferência Estatística é como obter um estimador de determinado parâmetro [91]. Um estimador é uma regra ou estratégia que usa dados para estimar parâmetros.

Jackman [159] afirma que numerosos métodos e suas variações são utilizados para estimar parâmetros de distribuição probabilística da população, tais como Estimadores de Momentos [91], Generalizado Método dos Momentos [147], L-Momentos [155], Mínimos Quadrados [91], Mínimos Quadrados Generalizados [90], Viável Mínimos Quadrados Generalizados [90], Máxima Verossimilhança [90], Quase-Verossimilhança [243], Inferência Bayesiana [198], Verossimilhança contrastado com Inferência Bayesiana [159], entre outros. Dois métodos bastante utilizados para estimação dos parâmetros são descritos como segue.

- Mínimos Quadrados

  O método dos Mínimos Quadrados ou *Ordinary Least Squares* (OLS) é baseado no Princípio dos Mínimos Quadrados, proposto por Gauss (1794) e publicado em 1809, relacionado a problemas de Astronomia e Física. Para calcular esse método, utiliza-se o Método de Gauss-Newton [91].

  Considerando os pontos (X_1, Y_1) (X_2, Y_2),...(X_N, Y_N), para o valor X_1, haverá uma diferença entre Y_1 e o valor correspondente determinado na função C. Essa diferença é apresentada por D_1 (desvio, erro ou resíduo), podendo ser negativo, positivo ou nulo. Da mesma forma em X_2,...X_N, obtêm-se os desvios D_2,...D_N (Figura 2.11). Assim, de todas as curvas que se ajustam a um conjunto de pontos, a que apresenta o mínimo valor de $D_1^2 + D_2^2 + ... + D_N^2$ é a melhor curva de ajustamento.

  Uma função que apresente esta propriedade, ou seja, que ajuste os dados no sentido dos mínimos quadrados, é denominada Curva de Mínimos Quadrados. Da mesma forma, uma reta com essa propriedade é denominada Reta de Mínimos Quadrados, assim como se for uma parábola, denomina-se Parábola dos Mínimos Quadrados [230].

[6]Baseada em trabalhos de J. Neyman, E. Pearson, R. Fisher, entre outros.

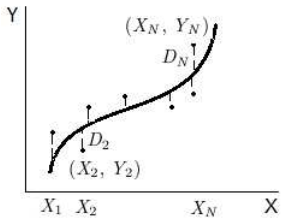


Figura 2.11: Curva de mínimos quadrados [230].

Conforme explicado em [230], essa definição é utilizada quando X é a variável independente e Y a dependente. Se as variáveis forem trocadas, resultam em curvas de mínimos quadrados diferentes.

O estimador *Ordinary Least Squares* - OLS é usado dentro do Modelo de Regressão Linear Clássico, mas não universalmente aplicável, segundo [90], pois um número de suposições deve ser satisfeita para este estimador, tais como: i. o modelo deve ser linear nos parâmetros; ii. resíduos são independentes e não relacionados; iii. variáveis independentes não são tão fortemente colineares; iv. variáveis independentes são medidas precisamente; v. o valor esperado dos resíduos é sempre zero; vi. os resíduos têm variância constante (variância homogênea); e vii. os resíduos são normalmente distribuídos.

- Máxima Verossimilhança

  Outro método popular de estimação proposto por R. A. Fisher (1912) e completo no ano de 1922, com a inclusão da expressão Verossimilhança[7], sendo um princípio onde a escolha do valor do parâmetro desconhecido maximiza a probabilidade de obter a amostra particular observada, tornando a amostra a mais provável [91]. Esse método escolhe os valores dos parâmetros mais consistentes com os dados, selecionando o valor do parâmetro que recebe a maior verossimilhança, dada uma amostra da população de interesse [90].

  A partir de uma amostra, é verificado no espaço paramétrico, quais os valores são mais indicados ou mais prováveis (verossímeis) para os parâmetros. Porém, em muitas situações, pode ser difícil ou até mesmo impossível de ser calculado, pois pode se tornar extremamente lento, devido à função de verossimilhança complexa, com número grande de parâmetros, sendo computacionalmente intensivo. Todavia, a tendência atual é propor modelos cada vez mais complexos para análise de conjuntos de dados em quase todas as áreas da ciência, pois o avanço tecnológico contribui para disseminar o uso desses métodos, conforme [120].

  Independente do método usado, após ter estimado os parâmetros do modelo, é necessário verificar se o mesmo, completamente derivado, de fato representa as séries temporais. Isto pode ser feito por considerar medidas da qualidade de ajuste do modelo (*goodness-of-fit*) [91].

[7]Verossímil é aquilo que é semelhante à verdade, provável e Verossimilhança é a qualidade ou caráter de verossímil. Fonte: Novo Dicionário Aurélio da Língua Portuguesa (2ed., 1986).

2.6.1.2 Análise Multivariada

Segundo Hair [144], qualquer análise que envolva mais de duas variáveis pode ser considerada uma Análise Multivariada. A Regressão Simples (com uma variável independente), é estendida para o caso multivariado quando incluir várias variáveis independentes (Regressão Múltipla). A análise multivariada inclui tanto técnicas de múltiplas variáveis quanto técnicas verdadeiramente multivariadas.

- Análise de Regressão Múltipla

  Uma extensão ao modelo de tendência linear é complementar ao índice de tempo t, com os valores de co-variáveis, podendo ser responsáveis por mudanças na variável resposta. Spiegel [230] cita que modelos de regressão múltipla, envolvendo mais de duas variáveis são tratados da mesma forma que com duas variáveis, onde a regressão (estimação) de uma variável (dependente) é feita a partir de duas ou mais de variáveis correlatas (independentes). O objetivo da análise de regressão múltipla, segundo Hair [144], é prever mudanças na variável dependente Y como resposta às mudanças nas variáveis independentes X. Nesse caso pode ser usado o estimador de Mínimos Quadrados. A representação de múltiplas co-variáveis em um modelo de regressão múltipla é dada pela Equação 2.19 [95]:

  $$Y_t = \beta_0 + \sum_{i=1}^{p} \beta_i x_{it} + \varepsilon_t = \mu_t + \varepsilon_t \tag{2.19}$$

  onde (t=1,...T) e x_{it} denota o valor da *ith* co-variável no tempo t.

  Quando todas as suposições são satisfeitas, a regressão múltipla representa todos os processos, afetando a quantidade de interesse em um simples modelo. Um exemplo em Chandler e Scott [95] é o ajuste de um modelo de regressão múltipla para dados originais (*raw data*), contendo co-variáveis representando a sazonalidade.

  Na Regressão Múltipla pode haver muitas variáveis a serem selecionadas para uso na equação de regressão, havendo vários métodos para seleção de variáveis, tais como de busca sequencial e processos combinatórios, ajudando na escolha do melhor modelo de regressão. Dentre as abordagens de busca sequencial, destacam-se [144, 115]: Estimação *Stepwise*, a qual inicia selecionando o melhor preditor da variável dependente e variáveis independentes adicionais são selecionadas considerando seu poder explicativo de forma incremental; Adição *Forward*, onde a modelagem é iniciada sem variáveis e depois são acrescentadas variáveis com base em sua contribuição na previsão; e Eliminação *Backward*, a qual inicia incluindo todas as variáveis independentes e elimina as que não oferecem contribuição significativa para a previsão. Dentre os processos combinatórios é citado a regressão em todos os possíveis subconjuntos.

- Análise dos Componentes Principais

  Dentre as técnicas multivariadas, no contexto da Análise Fatorial, destaca-se a Análise dos Componentes Principais (*Principal Component Analysis* - PCA), um método que permite analisar as inter-relações entre um grande número de variáveis e explicar essas variáveis considerando suas dimensões (fatores). O objetivo do método PCA é condensar as informações contidas em um grande número de variáveis originais, em um conjunto menor de variáveis estatística, representando os fatores

com menor perda de informações [144]. Como exemplos de métodos que utilizam PCA destacam-se o método de Regressão de Componente Principal (*Principal Component Regression*) que é usado para tratar o problema de multi-colinearidade, o qual ocorre quando qualquer variável independente é altamente correlacionada com um conjunto de outras variáveis independentes [144] na análise de regressão múltipla [177]; e a Análise do *Spectrum* Singular (SSA) [58], abordado na Seção 2.6.2.2.

2.6.2 Métodos de Estimação de Tendência Não-Paramétricos

Na modelagem de tendência de forma não-paramétrica, uma função suave de tendência é ajustada para os dados. Em geral, de acordo com Chandler e Scott [95], métodos de estimação de tendências paramétricos são baseados na suposição que a tendência apresenta uma forma matemática específica. Por exemplo, sendo linear ou que depende linearmente das co-variáveis. Segundo os autores, essa suposição nem sempre é adequada. Nesse caso, a tendência pode ser estimada usando um método não-paramétrico, onde tendências e relacionamentos são determinados a partir dos dados em si, melhor do que estar em conformidade com uma estrutura matemática pré-especificada.

Métodos de suavização podem ser usados para remoção do ruído das séries, fazendo uso de um método de médias móveis para suavizar o ruído branco (*white noise*) [223]. Todavia, métodos de suavização podem ser úteis para identificar certas características (padrões) nas séries, tais como tendência de longo prazo e componentes sazonais. Como forma de permitir maior flexibilidade na modelagem da tendência, a suposição de linearidade é removida, onde é possível definir um modelo de regressão não-paramétrico.

Dada a equação $Y_t = m(x_t) + \varepsilon_t$, onde (t=1,...,T), onde (ε_t) é uma sequência de ruído branco com variância $(\delta\sigma_2)$. Nesse caso a função de tendência paramétrica $(\beta_0 + \beta_1 x_t)$ é substituída por uma função arbitrária $m(x_t)$, cuja forma matemática não é especificada, definindo um modelo de regressão não-paramétrico com erros independentes. Para não considerar que a função de regressão m(.) seja totalmente arbitrária, assume-se que a mesma é suave, onde pequenas mudanças em x levam a pequenas mudanças em m(x) e, portanto, são associadas com pequenas mudanças em Y, em média. Assim, muitos métodos de regressão não-paramétricos são associados com alguma forma de suavização [95].

Na prática, para uma suavização, é necessário especificar um meio de suavizar uma matriz, Chandler e Scott [95] explicam que existem duas principais decisões nesse caso: i. Como calcular os valores resposta em cada vizinhança, ou seja, o tipo de suavização a ser feita e; ii. Qual tamanho da vizinhança é requerido, o qual é expresso em termos do parâmetro de suavização (*bandwidth*).

A seleção do parâmetro de suavização, ou seja, o tamanho da vizinhança, está intimamente relacionado ao problema de selecionar o grau em uma regressão polinomial ou, em selecionar variáveis em uma regressão múltipla. A Figura 2.12 mostra o efeito do parâmetro de suavização em funções de peso *Kernel* (*Kernel weight*), onde quanto maior a largura da banda, mais pontos obtêm peso diferente de zero, mas o peso de cada um é menor [62].

Os principais métodos de suavização usados na regressão não-paramétrica são [158]: suavização baseada em filtro (*smoothing filter*) ou suavização baseada em regressão local. Outra forma utiliza suavização *splines* [240], regressão *splines* [175], entre outros. De forma não-paramétrica, a tendência também pode ser identificada a partir das séries filtradas. Tais métodos são descritos na sequência. Para uma descrição mais detalhada

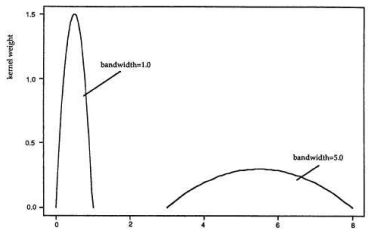


Figura 2.12: *Bandwidth* em funções de peso *kernel* [62].

sobre esses métodos, consultar suas respectivas fontes bibliográficas.

2.6.2.1 Análise de Regressão Não-Paramétrica

A análise de regressão não-paramétrica envolve os métodos Regressão Local, incluindo Regressão Polinomial Local, onde são descritos os métodos de suavização *Loess* e *Lowess* [102], Regressão *Kernel* [241] e Regressão Vizinho Mais Próximo. Também são abordados os métodos Suavização *Spline*, Regressão *Spline* e Suavização baseada em filtros digitais.

- Suavização baseada em Regressão Local (*Local Regression Based Smoothing*)

  Shumway e Stoffer [223] explicam que muitos métodos podem ser usados para suavizar os dados de séries temporais, baseados em métodos chamados *scatterplot smoothers*, resultando em $x_t = f_t + y_t$, onde f_t é uma função suave do tempo e y_t é um processo estacionário. Métodos de regressão moderna podem ser usados para ajustar suavizadores para os pares de pontos (t, x_t), onde a estimativa f_t é suavizada.

  Cleveland e Loader [103] descrevem princípios e métodos de suavização por regressão local e destacam quatro componentes básicos: a função peso (weight); a família paramétrica (ajustada localmente); o parâmetro de suavização e; o critério de ajuste. O modelo para regressão local: E(y_i) = f (x_i), i=1,...n, onde y_i são as observações de uma resposta e as d-tuplas x_i são observações de d variáveis independentes. A distribuição de y_i, incluindo as médias, f(x_i) são desconhecidas.

  Na prática, para modelar os dados, são feitas suposições sobre f e outros aspectos da distribuição de y_i. Uma suposição distribucional comum, por exemplo, é considerar que y_i tem uma variância constante. Para f, é suposto que a função pode ser aproximada localmente por um membro de uma classe paramétrica, frequentemente polinomial de determinado grau, chamado de localização paramétrica. A localização paramétrica é o aspecto fundamental que distingue métodos de regressão local a partir de outros métodos de suavização tais como suavização *spline* [240], regressão *spline* com seleção de nodos (*knots*) [132], entre outros métodos, embora essa noção é implícita nesses métodos de uma variedade de formas. O uso de métodos de regressão local para suavização é descrito na sequência [223].

  - Suavização *Loess* (*Loess Smoothing*)

    Um método mais complexo de suavização é *Lowess Smoothing* (*Locally Weighted Scatterplot Smoothing*), proposto por Cleveland [102]. Determinada proporção dos vizinhos mais próximos de x_t são incluídos em um esquema de peso, onde valores mais próximos de x_t no tempo obtêm mais peso. Uma regressão

ponderada (robusta) pode ser usada para prever x_t e obter a estimativa suavizada da função f_t. Quanto maior a fração dos vizinhos mais próximos incluídos, mais suave será a estimativa da função suavizada f_t [223].

Em [203], *loess* é citado como um método de modelagem moderno, construído usando métodos clássicos, tais como regressão de mínimos quadrados linear e não-linear. Ou seja, conforme citado, este método combina a simplicidade da regressão linear com a flexibilidade da regressão não-linear, ajustando simples modelos para subconjuntos localizados de dados, usando uma função que descreve a parte determinística da variação nos dados, ponto a ponto. Esse método não ajusta uma função global aos dados, somente para segmentos de dados.

Em cada ponto no conjunto de dados, um polinômio de baixo grau é ajustado para um subconjunto de dados, com valores de variáveis explanatórias perto do ponto cuja resposta é estimada. O ajuste polinomial é feito usando mínimos quadrados ponderados, estabelecendo mais peso aos pontos próximos do ponto cuja resposta está sendo estimada e menos peso para pontos mais distantes. O valor da função de regressão para o ponto é obtida pelo polinômio local, usando valores da variável explanatória para aquele ponto de dado. O ajuste *loess* está completo, após os valores da função de regressão terem sido computados, para cada um dos n pontos de dados. Tanto o grau do polinômio quanto os pesos são flexíveis. Os subconjuntos de dados usados para cada ajuste de mínimo quadrado ponderado em *loess* são determinados pelo algoritmo do próximo vizinho (*nearest neighbor*) [203].

O método *loess* apresenta dois parâmetros [158], o parâmetro de suavização, onde mais altos valores produzem ajustes mais suaves e, o grau da regressão polinomial. O parâmetro de suavização determina a quantidade dos dados usada para ajustar cada polinômio local, ou seja, este parâmetro controla a flexibilidade da função de regressão *loess*. Ajustes polinomiais locais para cada subconjunto de dados são quase sempre de primeiro (localmente linear) ou segundo grau (localmente quadrático). Com zero grau, *loess* se torna uma média móvel ponderada. Polinômios de mais alto grau não são indicados, pois esse método considera o ajuste de polinômios de baixo grau. Uma função peso ajusta mais pesos para os pontos de dados próximos ao ponto de estimação e menos peso aos pontos de dados que estão mais distantes deste. *Lowess* usa um polinômio linear (grau 1) e *loess* usa um polinômio quadrático (grau 2). Nesses métodos, o dado é modelado localmente, por meio da regressão de mínimos quadrados ponderados, onde os pesos dão mais importância para os pontos de dados locais, e a função peso tradicional usada é a função peso tricubo (*tricube weight function*), mas pode ser usada qualquer outra função que satisfaça as propriedades do método [203]. Esse método é adequado para um conjunto de dados grande, densamente amostrado.

Caso as séries apresentem *outliers*, os valores suavizados podem ficar distorcidos e não refletir o volume de pontos de dados vizinhos. Nesse caso, os dados podem ser suavizados usando um método robusto que não é influenciado por uma pequena quantidade de *outliers*. Os métodos *loess* e *lowess* apresentam uma versão robusta nesse sentido, apresentando cálculos adicionais de pesos robustos, resistentes a *outliers* [158, 203].

Como vantagens deste método citam-se [158, 203]: o mesmo não requer a

especificação de uma função para ajustar um modelo para todos os dados na amostra e o analista necessita fornecer somente o parâmetro de suavização e o grau do polinômio local. Trata-se de um método de suavização flexível, ideal para modelagem de sistemas complexos, para os quais não existem modelos teóricos. Como desvantagens do método *loess*, o mesmo é computacionalmente intensivo, necessitando de um conjunto de dados densamente amostrado, para produzir bons modelos e o analista necessita ter um conhecimento empírico da estrutura local do processo para fazer o ajuste local. Esse método não produz uma função de regressão, representada por uma fórmula matemática, segundo [158, 203].

– Suavização baseada em Regressão *Kernel* (*Kernel Smoothing*)
A suavização *Kernel* [241] é um método de médias móveis que usa uma função peso (*weight*) ou *kernel* para calcular a média das observações. Este método geralmente utiliza o estimador *Nadaraya-Watson* [201]. Quanto mais amplo for o parâmetro de suavização (*bandwidth*), mais suave é o resultado [223]. O método *kernel* tem um parâmetro de suavização que controla o grau de pontos de influência do ajuste local. Se esse parâmetro for pequeno, a curva será ondulada, porque a estimativa depende de pontos próximos a determinado ponto. Valores altos desse parâmetro significam que uma linha será ajustada para o conjunto de dados inteiro, conforme Isnanto [158]. A Figura 2.13 apresenta um exemplo de suavização *kernel* em dados de mortalidade cardiovascular na cidade de Los Angeles sob o período de dez anos (1970-1979) [223].

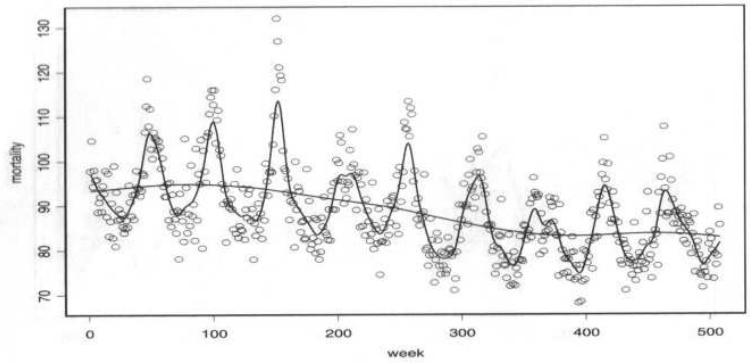


Figura 2.13: *Kernel smoothing* [223].

A suavização *kernel* é um caso especial de regressão local [103], um método *kernel* consiste na escolha da família paramétrica consistindo em funções constantes. Segundo Isnanto [158], este método é similar a *loess*, onde o ajuste ocorre de modo local, chamados estimadores *kernel* polinomial local. Existem vários estimadores, como por exemplo, *Nadaraya-Watson* [201], *Priestley-Chao* e estimador *Kernel Local Linear*. Com métodos de suavização *Kernel*, problemas podem surgir nos limites da amostra, isso porque a janela *kernel* nos limites tem valores ausentes. Nesse caso, existem pesos associados, mas nenhum dado é associado a estes. O estimador local linear se comporta bem nesses casos e no caso do estimador *Nadaraya-Watson*, é necessário o uso de *kernels* modificados [158].

– Suavização baseada em Regressão Vizinho Mais Próximo (*Nearest Neighbor Smoothing*)

Uma abordagem para suavização das séries temporais é baseada na regressão do vizinho mais próximo (*Nearest Neighbor Regression*), a qual faz uma regressão linear dos k mais próximos vizinhos, em que é usado os dados $x_{t-k/2}, ..., x_t, ...x_{t+k/2}$ para predizer x_t, usando regressão linear [223]. Segundo Chandler e Scott [95], esse método pode ser particularmente útil para séries temporais irregularmente espaçadas. A Figura 2.14 apresenta um exemplo dos métodos Vizinho Mais Próximo e *Lowess* em dados de mortalidade [223].

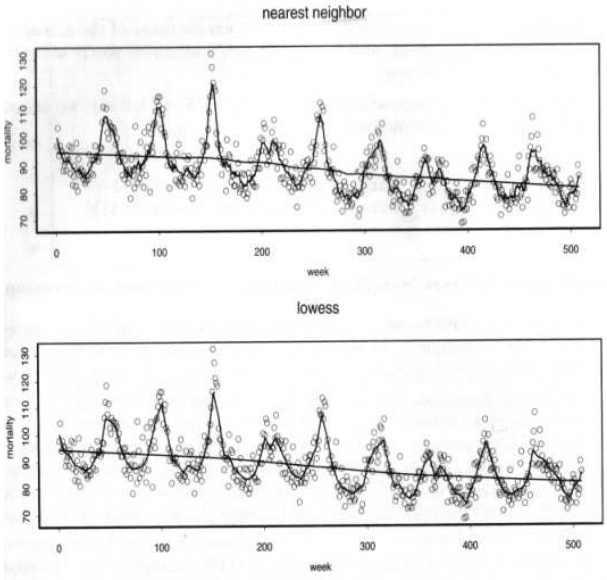


Figura 2.14: *Nearest neighbor e lowess* [223].

Altman [62] cita que estimadores de regressão *kernel* [201] e vizinho mais próximo [71] são versões locais de estimadores de localização univariada. São simples extensões de estimadores de localização univariada ordinários, sendo poderosas ferramentas analíticas de dados, tanto como técnicas autônomas como suplementos para análises paramétricas. Estimadores do vizinho mais próximo usam bandas (quando divide o *scatterplot* em bandas individuais e computa-se um estimador de localização em cada banda) de tamanho de amostra constante. Geralmente, a vizinhança é escolhida de forma que um número igual de pontos são selecionados a partir de cada lado do ponto estimado.

- Regressão *Spline* (*Spline Regression*)

Muitas variações de *splines* cúbicos (base usada para representar a função de regressão) estão disponíveis para regressão não-paramétrica. Por exemplo, em uma regressão usando *spline* cúbica, os segmentos são definidos não pelos valores das co-variáveis individuais, mas por um conjunto K $\ll$ T nodos, escolhidos para cobrir o intervalo inteiro das co-variáveis [95].

Em uma regressão *spline*, são escolhidos conjuntos de nodos muito menores que os pontos de dados e um conjunto de funções base, abrangendo um conjunto de

polinômios por partes, satisfazendo continuidade e restrições de suavidade. A função *spline* linear (por partes) é construída por regredir os dados nessas funções base. Funções de base *spline* linear têm derivativas descontínuas, onde o ajuste resultante pode ter uma aparência irregular. Por isso é mais comum usar *splines* cúbicos (por partes), com as funções base tendo duas derivativas contínuas [175]. A Figura 2.15 apresenta um exemplo de regressão paramétrica não-linear e regressão *spline*, considerando a altura de uma pessoa como uma função da idade.

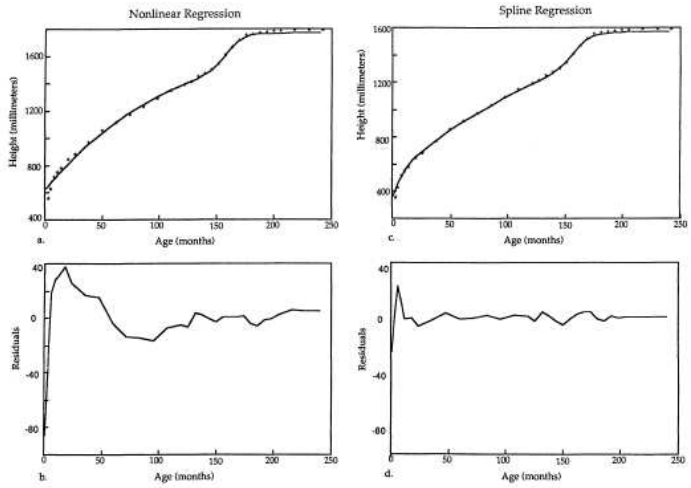


Figura 2.15: Regressão paramétrica não-linear e regressão *spline* e resíduo [62].

- Suavização *Spline* (*Smoothing Spline*)

  Um algoritmo de suavização *spline* cúbico para estimação de tendências em séries temporais é proposto em [210], descrevendo uma função *spline* como uma curva ajustada, a partir de segmentos polinomiais, sujeitos a condições de continuidade em suas junções. *Smoothing spline* minimiza o compromisso entre o ajuste e o grau da função de suavidade [210]. A Figura 2.16 apresenta um ajuste de suavização *spline* em dados de mortalidade cardiovascular semanais [223].

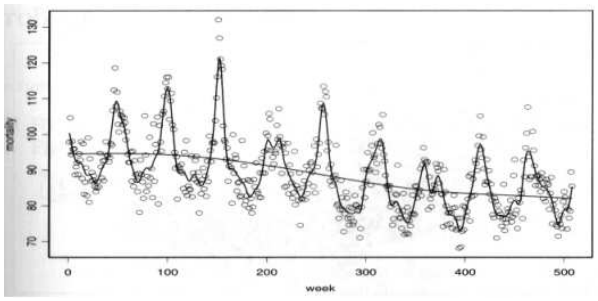


Figura 2.16: Ajuste *Smoothing spline* em dados de mortalidade [223].

Segundo Shumway e Stoffer [223], uma extensão da regressão polinomial é dividir o tempo t em k intervalos, onde os valores dos intervalos são chamados nodos (*knots*). Em cada intervalo, é ajustada uma regressão. Se o grau da função de regressão é três, é ajustada uma função *spline* cúbica.

Suavização *spline* [240], incluindo suavização *spline* cúbica, conforme Chandler e Scott [95], caracterizam uma forma de desenvolvimento de um suavizador linear,

onde é construída uma estimativa da função de regressão não-conhecida, a partir de uma coleção (ou base) de funções mais simples e, melhor do que minimizar a soma dos quadrados, é usado uma penalizada soma de quadrados. Um parâmetro de suavização é usado da mesma forma que em um suavizador linear local. Trata-se de uma abordagem de suavização diferente, feita através de otimização de um critério de mínimos quadrados penalizado, onde a penalidade é especificada constante. Esse critério está entre a fidelidade aos dados, medida pela residual soma de quadrados *versus* a robustez da função média, medida pelo termo de penalidade. A solução para este problema de otimização é um polinômio ajustado por partes (*piecewise*) ou por função *spline*. Métodos mínimos quadrados penalizados são também conhecidos como suavização *spline*, segundo [175].

Loader [175] explica que um ajuste *smoothing spline* pode ser muito similar à regressão local, considerando uma simples variável preditora e parâmetros de suavização comparavelmente escolhidos. Por outro lado, métodos *kernel* se esforçam em produzir resultados aceitáveis, mesmo em conjuntos de dados relativamente simples.

Segundo Racine [214], regressão *spline* difere de várias formas de *smoothing splines* [240], os quais são dois métodos populares de estimação não-paramétrica. A diferença fundamental é que *smoothing splines* explicitamente penalizam irregularidades e usam os pontos de dados em si como potenciais *knots* e regressão *spline* usa *knots* em pontos equidistantes. Conforme Chandler e Scott [95], *penalized splines* (*smoothing splines*) são uma alternativa à regressão *spline* a qual restringe os coeficientes *spline* em segmentos adjacentes para serem similares uns aos outros de alguma forma.

- Suavização baseada em Filtro de Suavização (*Smoothing Filter based Smoothing*)

  Considerando a suavização linear, o filtro de média móvel pode ser usado para gerar séries suavizadas, sendo usado pesos iguais para as médias, contribuindo para identificar a tendência das séries temporais [223]. Segundo [95], a média móvel é feita com a média das observações dentro da vizinhança de cada ponto de interesse. A Figura 2.17 apresenta um exemplo de uso de média móvel de 5-semanas e 53-semanas para séries de mortalidade cardiovascular semanais.

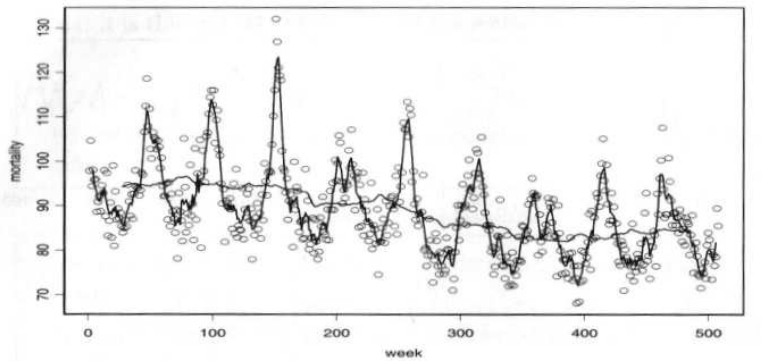


Figura 2.17: Média Móvel em séries de mortalidade cardiovascular semanais [223].

Isnanto [158] explica que ao aplicar uma média móvel em dada série, o valor para um dado período de tempo é substituído pela média daquele valor e os valores

para algum número de antecedentes e precedentes períodos de tempo. O método de suavização baseado em *smoothing filter* é uma das primeiras técnicas usadas para suavização, onde o filtro suaviza dados substituindo cada ponto de dado com a média dos pontos de dados vizinhos, definidos dentro de um intervalo (*span*). Em linguagem de frequência, este método é equivalente a um filtro passa-baixa frequência [94], ou seja, permite a passagem da tendência que é de baixa frequência e elimina ou atenua componentes de alta frequência como o ruído. Conforme explicado, algumas regras são aplicadas nesse método de suavização, onde o intervalo deve ser ímpar; os pontos de dados a serem suavizados devem estar no centro do intervalo; o intervalo é ajustado para pontos de dados que não podem acomodar um número especificado de vizinhos de cada lado e; os pontos finais não são suavizados porque um intervalo não pode ser definido.

Um exemplo de método de suavização baseada em filtro, usando médias móveis, é o método Savitzky-Golay [158], o qual é visto como um método de média móvel generalizada. Os coeficientes do filtro são derivados a partir de um ajuste linear de mínimos quadrados não-ponderado usando um polinômio de determinado grau. Esse método é conhecido como um filtro digital de suavização polinomial ou um filtro de suavização de mínimos quadrados. Um alto grau da função polinomial possibilita alcançar mais alto grau de suavização, sem atenuar as características dos dados. Para uso deste método, o intervalo das observações deve ser ímpar; o grau polinomial deve ser menor do que o intervalo; e os pontos de dados não requerem um espaçamento uniforme. Entretanto, segundo os autores, esse filtro é menos adequado para rejeitar o ruído do que o filtro de médias móveis.

Filtros digitais podem ser usados para estimação e/ou remoção de tendência de forma não-paramétrica, assim como podem ser usados para remoção de ruído. Kim et al [164] explicam que o problema de filtrar a tendência (*trend filtering* ou *trend estimation*) está relacionado a muitas áreas e que muitos métodos de filtro têm sido propostos, tais como os filtros lineares Hodrick-Prescott, médias móveis, suavização exponencial, filtros passa-banda, suavização *splines*, *detrending* via filtros *rational square-waves* e os filtros não-lineares como o filtro de medianas móveis. Filtrar é um método para estimar a tendência, a partir do histórico inteiro de observações.

O objetivo de filtrar uma série temporal é extrair estruturas de interesse, sendo um recurso útil na fase exploratória dos dados. Após suavizar as séries para visualizar quaisquer tendências aparentes mais facilmente, é possível aplicar testes para verificar se as tendências nos dados suavizados são verdadeiras [95].

Filtrar os dados é um processo onde as séries são separadas em suas partes-componentes, sendo um método frequentemente usado para remoção de impuridades. Considerando uma série temporal apresentando tendência e alguma variação irregular (a impuridade), idealmente, o efeito de suavizar é a remoção da variação irregular, resultando na tendência. O uso de um filtro linear separa uma dada série temporal no componente tendência e no componente irregular (ou flutuação). Nesse caso, a série considerada consiste de tendência mais variação irregular. Na linguagem de frequências, a tendência contribui para baixas frequências, ao passo que a variação irregular contribui para altas frequências [95] (Figura 2.18).

A escolha de um filtro, segundo Chandler e Scott [95] depende do que se deseja extrair das séries. Ao verificar que a tendência é linear, então deve ser usado um filtro

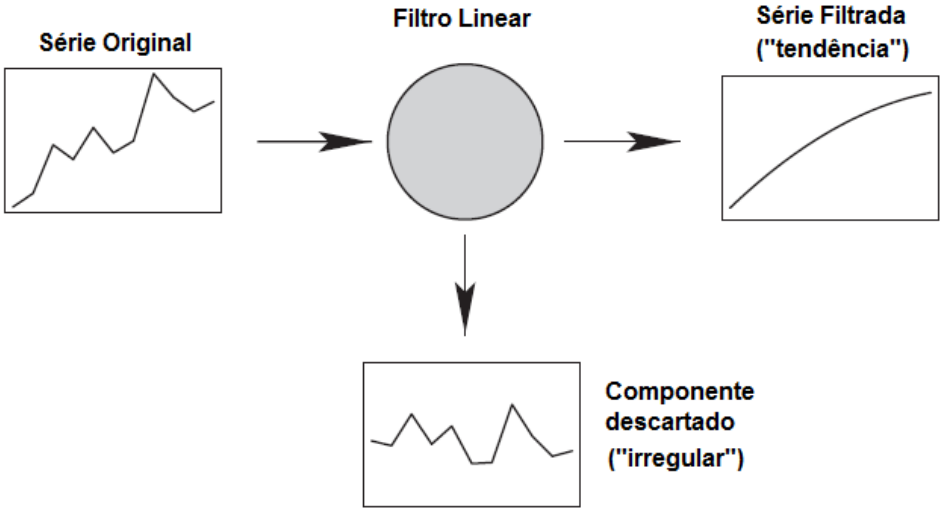


Figura 2.18: Filtro linear [95] (tradução).

pelo qual a tendência linear passará. Da mesma forma, ao considerar a tendência como polinomial, deve-se usar um filtro que permita a passagem dos mesmos. No caso da estimação da tendência, é preciso um filtro onde somente componentes de baixas frequências passem. Como anteriormente citado, um número infinito de filtros pode ser usado, os quais são descritos na seção 2.6.2.2.

Quanto à escolha do parâmetro de suavização (*bandwidth*), Chandler e Scott [95] explicam que quando as séries temporais são suavizadas para visualizar tendências, este resultado depende da escolha precisa de pesos. A regressão não-paramétrica depende do parâmetro de suavização escolhido. Pollock [210] comenta que a dificuldade está em encontrar um critério objetivo para escolher o valor do parâmetro que está entre a suavidade da curva e sua proximidade com os pontos de dados. Em suavizadores local linear e *spline*, aumentar indefinidamente o parâmetro de suavização implica que a função de regressão estimada será uma linha reta. Por outro lado, ao diminuir este parâmetro, o suavizador tende a interpolar os dados. Como filtros podem ser usados para extrair componentes de interesse das séries, considerações similares podem ser usadas para escolha do parâmetro de suavização neste contexto, segundo Chandler e Scott [95].

Existem vários métodos disponíveis para escolha do parâmetro de suavização, tais como [95]: Cross-Validação (*Cross-Validation*) e Generalizada Cross-Validação (*Generalized Cross-Validation*), onde a idéia geral é omitir cada ponto de dado para ajustar o modelo usando um intervalo de diferentes parâmetros de suavização e usar os modelos ajustados para prever a variável resposta nos pontos de dados omitidos. Tais métodos são considerados duvidosos na presença de auto-correlação, pois é difícil distinguir entre tendências suaves e forte auto-correlação. Uma alternativa para Cross-Validação é calcular os efetivos graus de liberdade (*degress of freedom*) do modelo ajustado. Os graus de liberdade correspondem ao número dos coeficientes de regressão estimados e os graus de liberdade residuais correspondem ao número de pontos de dados independentes remanescentes.

Uma alternativa aos métodos formais é o uso de métodos gráficos, comumente utilizados para escolher um apropriado parâmetro de suavização [95]. Nesse caso, o analista faz um julgamento baseado em uma inspeção visual dos dados e a função de regressão estimada,

levando em consideração o conhecimento do sistema físico para julgar se a estimativa específica da função de regressão é adequada. Tal abordagem é considerada subjetiva, mas pode ser útil para repetir análises, usando um intervalo de valores desses parâmetros que contribuem para a tomada de decisão se o parâmetro de suavização é adequado.

2.6.2.2 Filtros

Segundo Alexandrov et al [59], além da abordagem de especificação de um modelo para a tendência, outra abordagem é visualizar a tendência como a saída de um filtro linear aplicado aos dados. Um filtro é definido como um mecanismo que permite passar componentes, em uma determinada faixa de frequência [198]. Segundo a literatura, filtros podem ser classificados de diversas formas, como a implementação, linearidade, domínio, projeto, entre outras formas.

Filtros podem ser classificados conforme sua implementação, usando convolução ou recursão [223, 70]. O filtro convolução apresenta uma função resposta de impulso do filtro, a qual é determinística e depende da estrutura do sistema, porém é independente da forma da entrada. A função de transferência do filtro é dada pela Transformada de *Fourier* da função resposta de impulso. Filtros recursivos produzem uma resposta desejada, distorcendo as fases de frequência na entrada [223].

Filtros digitais podem ser classificados como lineares ou não-lineares [183]. A Figura 2.19 mostra a notação convencional para sistemas lineares de uma dimensão [94]. A entrada para o sistema é uma função de uma variável e é produzida uma resposta a partir do sistema que é outra função da mesma variável. Qualquer sistema que não obedece essa restrição é não-linear. A análise de sistemas não-lineares tem produzido resultados úteis em vários domínios, mas sua análise é consideravelmente mais complexa do que sistemas lineares.

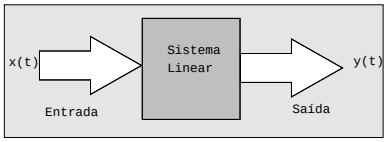


Figura 2.19: Sistema linear [94].

Shumway e Stoffer [223] citam que filtros lineares podem ser usados para extrair sinais, a partir das séries. Um filtro linear modifica as características espectrais de uma série, sendo um tópico relevante na análise de séries temporais. Um exemplo de um filtro linear é o de médias móveis.

Meinl [183] descreve o uso de filtros lineares e não-lineares não-paramétricos (que não requerem a especificação de um modelo) usados na extração de tendências. Filtros lineares são conhecidos por apresentar uma tendência suave e os filtros não-lineares se destacam por preservar características padrão em uma série, especialmente *jumps*. Ambos os filtros requerem a especificação de pesos (*weights*) e parâmetros de calibração. As classes de filtros lineares e não-lineares são separadas de acordo com, se um filtro pode ser descrito por uma função de transferência, a qual faz uma estrita divisão entre essas duas classes de filtros. Essa tese segue essa classificação, conforme explicado em [183].

Filtros lineares são os mais comuns e conhecidos, usados na extração de tendências e remoção do ruído aditivo. Quanto às funções de transferência, filtros lineares podem ser considerados no domínio do tempo ou da frequência. No primeiro caso, uma série

e sua respectiva saída filtrada evolui sobre o tempo [183]. O projeto (*design*) de filtros finitos, com uma resposta de frequência especificada, requer experimentação com várias funções resposta de frequência. O *design* dos filtros lineares envolve filtros passa-alto, passa-baixo, passa-banda e pára-banda (rejeita-banda), conforme a Figura 2.20, os quais são explicados na sequência [223, 198].

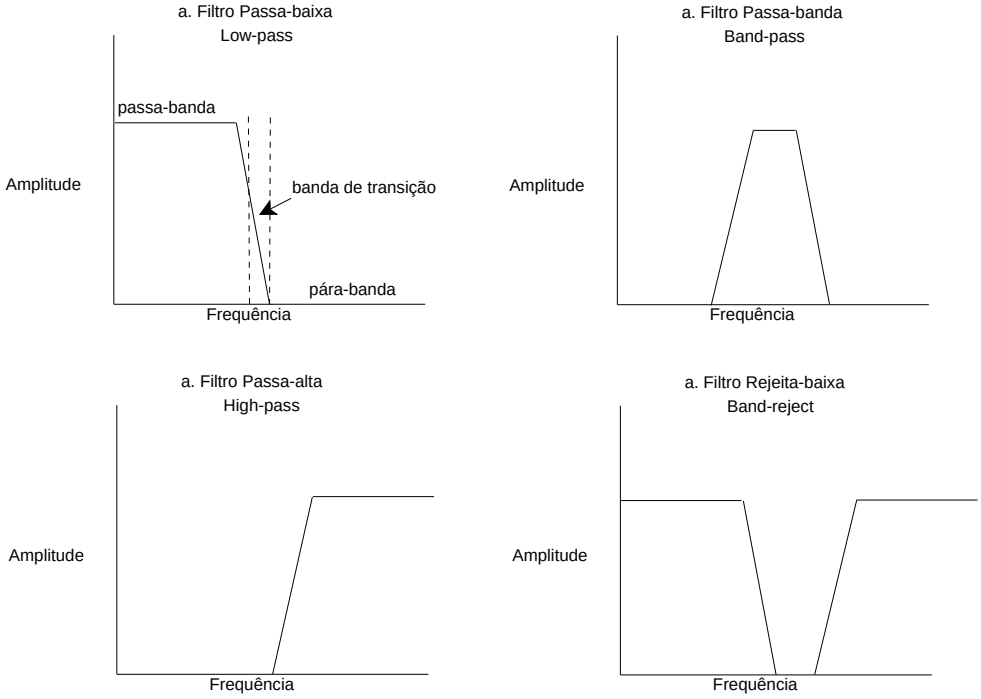


Figura 2.20: Passa banda filtros [228] (tradução).

O filtro passa-alto (*high-pass*) deixa passar (mantém) componentes com frequências altas e elimina (ou atenua) componentes de baixa frequência. Alguns filtros passa-alto são:

- Filtro Diferença: onde $\mathrm{Y}_t = (1-B)X_t = X_t - X_{t-1}$. Esse filtro elimina componentes de baixa frequência como a tendência e deixa passar o ruído (componente de alta frequência), conforme a Figura 2.21.

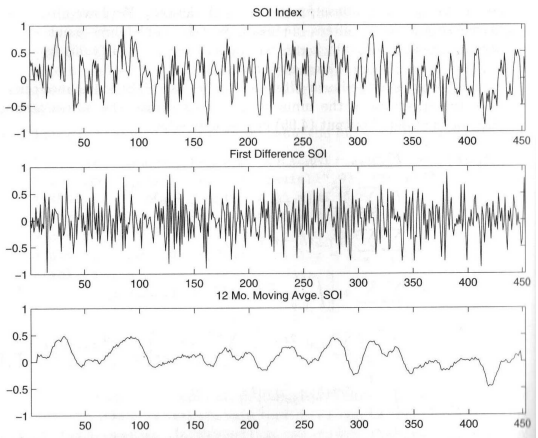


Figura 2.21: Primeira diferença e média móvel [223].

A diferenciação é uma maneira de remover a não-estacionariedade, a partir das séries temporais [223, 184]. Uma série temporal não-estacionária na média, ou seja apresenta tendência na média, pode se tornar estacionária por meio da Primeira Diferença [198]. Esse processo subtrai por meio de um operador de defasagem (*lag*) uma observação no instante de tempo t menos uma observação t *lags* anteriores, ou seja, a diferença do valor das séries nos tempos t e *t-1*: $w_t = x_t - x_{t-1}$, onde x_t é série temporal original, w_t é a série primeiro diferenciada (Figura 2.21).

Se a série é não-estacionária, não na média, mas na taxa de variação da média, a estacionariedade por ser obtida pela segunda diferença: u_t=w_t-w_{t-1} e, assim, sucessivamente, onde ordens mais altas de diferenciação podem ser aplicadas. Em situações normais, é suficiente uma ou duas diferenças para que a série se torne estacionária, ou seja, livre de tendências. Entretanto, a diferenciação pode ser problemática onde faltam dados originais nas séries diferenciadas [193].

- Filtro Diferença Sazonal: quando (S = 12), é dado por: $Y_t = (1 - B^{12})X_t = X_t - X_{t-12}$, o qual elimina as frequências correspondentes ao componente sazonal e seus harmônicos [198].

- Filtro Diferença Fracional: uma alternativa para o filtro Diferença, com uma operação menos severa, mas ainda assim, assumindo estacionariedade nas séries, é o filtro Diferença Fracional (*Fractional Differencing Filter*), o qual estende a noção do operador diferença para potências fracionais no intervalo ($-.5 < d < .5$), definindo um processo estacionário. Este filtro é, frequentemente, usado para séries temporais ambientais na hidrologia, conforme [223].

O filtro passa-baixo (*low-pass*) deixa passar (mantém) componentes com frequências baixas e atenua (reduz) componente com frequências altas. O filtro de médias móveis simétrico diminui a variabilidade (ruído) das séries, conforme a Figura 2.21. O filtro passa-banda (*band-pass*) mantém frequências em bandas específicas. Por exemplo, filtros de ajuste sazonal podem ser usados para rejeitar frequências sazonais, enquanto mantém ambas as frequências altas e baixas, inclusive a tendência. O *design* do filtro linear pára-banda (*band-stop*) envolve a saída do domínio da frequência, a qual é localizada em determinado intervalo de frequência.

Filtros podem ser aplicados de modo sequencial. Para exemplificar, pode ser aplicado um filtro Diferença, seguido por um filtro Diferença Sazonal, em uma dada série para eliminar, respectivamente, os componentes de baixa frequência, ou seja, a tendência e os componentes com frequências correspondentes a determinado período de sazonalidade e suas frações, conforme Morettin e Toloi [198].

A escolha de um filtro depende do objetivo da análise. Filtros também podem ser usados para correção do componente de alta frequência das séries temporais que é o ruído. Esse processo pode ocorrer durante a fase de pré-processamento e antes da estimação e extração de tendências em si. O objetivo é separar o ruído das séries temporais, objetivando sua redução ou remoção. Da mesma forma como o componente tendência, a definição de ruído é imprecisa pois, para determinados processos físicos, é difícil defini-lo de forma clara.

Segundo Meinl [183], a remoção de ruído (variações de curto prazo) a partir das séries não necessariamente leva a um sinal suave, pois a saída pode ainda conter sazonalidades

de curto prazo. Métodos usados para esse fim estão relacionados com a tarefa de correção das séries a partir de irregularidades.

Segundo Bendat e Piersol [70], métodos de suavização clássicos, interpolação, extrapolação, diferenciação e integração são exemplos de filtros FIR de resposta de impulso finito. Na sequência são apresentados exemplos de filtros que podem ser usados tanto para *detrending* quanto para extração de ruído, nesse último caso usando filtros de suavização de baixa frequência, conforme [59, 183].

- *Mean*: considerado o filtro linear mais conhecido, todos os pesos são uniformemente distribuídos. Filtros *kernel* são mais gerais que estes, os quais utilizam uma função *kernel* (*weight*), onde os valores do *kernel* podem corresponder a funções *kernel Uniform*, *Epanechnikov*, *Biweight* e *Triweight*.

- *Henderson*: conhecidos por minimizar a suavização em relação ao terceiro grau polinomial dentro do período do filtro. O problema de minimização é considerado, onde pesos simétricos são escolhidos para minimizar a soma de quadrados de suas terceiras diferenças, sendo o critério de suavização.

- *Hodrick-Prescott (HP)*: segue a abordagem da suavização *spline* cúbico. Apesar de não usar uma base *spline* para aproximação, pode ser visto como uma formulação discretizada dessa forma de suavização. Nesse filtro, uma dada série é a soma de um componente de crescimento T e um componente cíclico C, onde X = T + C. A medida de suavidade da tendência T é a soma dos quadrados de sua segunda diferença. Os desvios a partir de T são C e o *framework* conceitual, é que sob longos períodos de tempo, sua média é próxima de zero.

- *Rational Square-Waves*: um filtro de frequência seletiva, utilizando uma função racional (*rational*), uma razão de polinômios, embutida em um número limitado de coeficientes e cuja resposta de frequência é uma aproximação para uma onda quadrada (*square wave*) [211].

- *Binomial*: pode ser computado por repetida convolução da sequência de pesos, correspondendo a igual probabilidade de sucesso ou falha para uma distribuição binomial [184].

- *Gaussian*: configura pesos iguais para as ordenadas de uma apropriada função de densidade de probabilidade Gaussiana, ou Normal. Este filtro é, particularmente, conveniente pois o desvio padrão da distribuição Gaussiana apropriada pode ser especificada em termos de 50 porcento da resposta de frequência do filtro [184]. Este filtro pode ser projetado como um Gaussiano passa-alto.

- *Henderson*: são conhecidos por suavizar séries temporais, com um polinômio de terceiro grau, dentro de um período de dados [59].

- *Moving Average*: suaviza as séries porque retém as mais baixas ou mais lentas frequências e tende a atenuar as mais altas frequências [223].

- *Savitzky-Golay*: faz uma regressão polinomial local de determinado grau nas séries, para determinar o valor suavizado em cada ponto [5].

- *Smoothing Spline*: esse filtro de suavização *spline* faz a computação das derivativas para dados igualmente espaçados [125].

Além desses filtros, existem os filtros de resposta de impulso infinito, implementados usando recursão. Um filtro digital recursivo (IIR) é um tipo de filtro, onde a saída das séries temporais é gerada, não somente usando uma soma finita de termos de entrada, mas usando saídas anteriores como termos de entrada (um procedimento chamado retorno (*feedback*) [70].

Os filtros IIR clássicos são: *Butterworth*, *Chebyshev* Tipo I e II, e *Elíptico*. Esses filtros podem ser usados em ambos os domínios analógico e digital e em configurações de intervalos de frequência como passa-baixa (*lowpass*), passa-alta (*highpass*), passa-banda (*bandpass*), e rejeita-banda (*bandstop*) [17].

Embora a maioria dos filtros não lineares buscam apresentar uma tendência suave, Meinl [183] explica que na região ao redor dos *jumps*, filtros não-lineares falham em apresentar uma tendência suave, tão precisamente quanto um simples filtro linear, como o filtro *Mean* fornece. Outra questão é a ausência de controle de frequência, pois filtros não-lineares não regulam a saída filtrada em termos de passa-bandas de frequências, como filtros lineares. Isso porque tratam componentes das séries tais como ruído e *jumps*, mesmo se estes estão localizados perto do mesmo intervalo de frequência.

Filtros não-lineares não são somente aplicados na análise de séries temporais, sendo também usados para extração do ruído (*denoising*) de sinais de duas dimensões, especificamente imagens. Se um valor extremo (*outlier*) ou um salto aleatório (*jump*) estiver presente em uma janela de filtro de médias móveis ponderada, um peso é declarado para os pontos de dados de *outliers* ou nos pontos antes ou após os *jumps*. Nesse caso, filtros não-lineares tentam evitar isso usando uma abordagem diferente. Por exemplo, considerando um simples valor (ao invés de múltiplos ponderados), o qual é selecionado a partir de uma permutação ordenada (por exemplo, rankeada) dos valores originais localizados na janela do filtro, segundo [183].

Jumps e outras mudanças repentinas (*sudden changes*) são usualmente bem capturadas por filtros não-lineares, apesar de que esses filtros não apresentam uma tendência suave, uma vez que qualquer ruído, com altas amplitudes esteja presente. Os filtros não-lineares fazem operações tais como mínimo, máximo e mediana sobre uma vizinhança. Alguns dos principais filtros não-lineares são listados a seguir, segundo [183, 212, 94].

- *Median*: filtro de suavização de ruído não-linear.
- *Trimmed Mean*: se assemelha a um filtro da média, com a diferença que os valores extremos das séries ordenadas são cortados (*trimmed*), ou seja, descartam todas as amostras a partir das séries ordenadas que estão distantes conforme alguma medida. A ideia é rejeitar os mais prováveis *outliers* (alguns dos muito pequenos valores e os valores muito grandes).
- *Winsorized Mean*: é uma modificação do *Trimmed Mean*, onde não são descartados valores extremos das séries ordenadas mas, ao invés, estes são substituídos por valores próximos em uma unidade. Os menores valores dentro da janela são substituídos por x(r+1) e os maiores valores dentro da janela são substituídos por x(N-s).
- *K-Nearest Neighbor*: a saída é dada pela média de K, $1 \leqslant K \leqslant$N, amostras cujos valores estão próximos do valor central x* dentro da janela do filtro.
- *L-Filters (Order Statistics)*: *L-Filters* são também chamados filtros *Order Statistics*,

os quais executam operadores entre uma operação pura não-linear (*ordering*) e uma operação pura (*weighting*).

- *Ranked Order Statistic*: podem ser usados em situações onde a distribuição do ruído não é simétrica. Por exemplo, onde existem mais impulsos positivos do que negativos.

- *Weighted Order Statistic*: um filtro *rth ranked order statistic* é dado por considerar $X(_r)$ como saída do filtro. Exemplos são a operação mediana, o máximo (r=N) e o mínimo (r=1). Isso pode ser combinado com pesos.

- *Weighted Median*: é possível enfatizar as amostras que por alguma razão são supostas serem mais confiáveis. Por exemplo, a amostra central x*, onde a ênfase é obtida por um peso maior.

- *M-Filter*: uso de M-estimadores que são generalizações dos estimadores da máxima verossimilhança (*maximum likelihood*).

- *R-Filter*: uso de R-estimadores, conhecidos como filtros robustos.

- *Hybrid*: outra abordagem de filtros não-lineares consiste de filtros cascata (*cascade*), ou seja, consideram a aplicação repetida de diferentes filtros nas respectivas saídas. Um exemplo é o filtro *median hybrid* que combina uma operação de filtro linear com uma operação de ordenação mediana.

- *Selective*: nesse caso existe uma troca entre diferentes regras de saída, dependendo de alguma regra de seleção.

Existem outras formas de decomposição das séries, as quais além de estimar a tendência, fazem também sua extração. Por exemplo, para o filtro adaptativo aos dados Análise do Espectro Singular (SSA) são propostos vários métodos para extração de tendências, conforme [58]. Outros métodos como Modo de Decomposição Empírico (EMD) [156] também podem ser usados para extração de tendências. Esses métodos são descritos na sequência.

- *Empirical Mode Decomposition* (EMD) [156, 161]: é um método empírico para análise de séries temporais reais, no domínio tempo-frequência, o qual é auto-adaptativo, refletindo as características intrínsecas das séries temporais. EMD decompõe as séries temporais originais em modos oscilatórios, denominados funções de modo intrínseco (IMFs) e um termo residual, de acordo com a frequência das séries temporais. As IMFs são usadas para distinguir entre flutuações e tendências. Conforme Li et al [172], IMFs de ordens mais baixas (*lower-order*) capturam modos de oscilação mais rápidos e IMFs de ordens mais altas (*high-order*) e o resíduo, representam modos de oscilação mais lentos. Assim, IMFs refletem a característica intrínseca das séries temporais e o termo residual é a tendência das séries temporais.

  Wu et al [248] discutem que a maioria dos métodos para *detrending* definem tendência como uma abordagem extrínseca, como no caso de haver formas funcionais pré-selecionadas para a tendência, apresentando uma definição de tendência como intrínseca e adaptativa, usando dados climáticos. Nesse caso, a tendência não é nem linear nem

quadrática, é adaptativa aos dados. O método EMD permite uma operação de *detrending* nos dados e a determinação da variabilidade, acerca da linha da tendência, em processos não-estacionários e não-lineares.

A definição da tendência é aplicada a séries temporais de anomalias anuais de temperatura do ar da superfície global (GSTA - *Annual Global Surface Air Temperature Anomaly*). A Figura 2.22 mostra os dados GSTA da anomalia da temperatura global de 1856 a 2003 e a Figura 2.23 mostra as médias (linhas pretas) e desvio-padrão (linhas cinzas) de IMFs a partir de dez diferentes conjuntos.

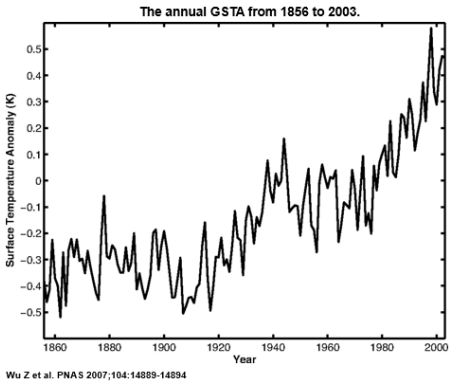


Figura 2.22: Dados anuais referentes à anomalia da temperatura global (1856 a 2003) [248].

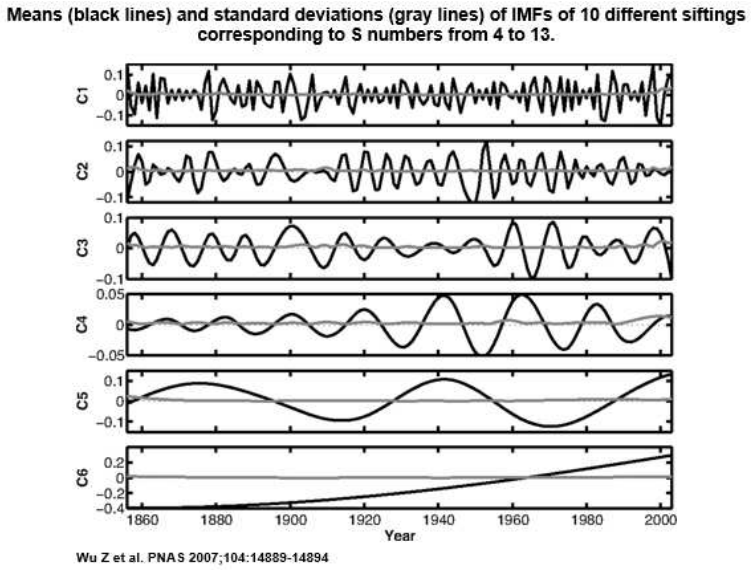


Figura 2.23: Média e desvio-padrão de funções de modo intrínsecas de dez diferentes conjuntos [248].

A Figura 2.24 apresenta uma comparação entre diferentes ajustes de tendência, onde os dados são a linha preta fina e suas tendências (tendência linear, linha cinza fina; tendência adaptativa global, linha preta grossa (componente residual C6); e tendência multidecadal (soma de C5 e C6), linha cinza grossa. A tendência linear é citada como a mais pobre, a tendência adaptativa global determinada de forma intrínseca mostra grande melhoria em relação à tendência linear e a tendência

multidecadal captura a variabilidade significativa e mudanças associadas com os dados, mostrando um ajuste ainda maior.

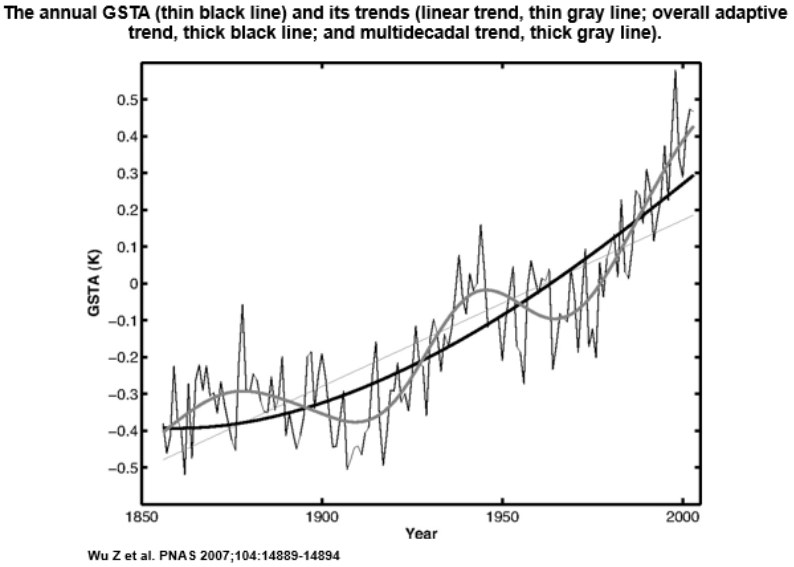


Figura 2.24: Dados anuais referentes à anomalia da temperatura global e tendências linear, adaptativa global e multidecadal [248].

Portanto, apesar de haver o modelo não-linear, onde funções extrínsecas tais como exponencial, *power law* e hiperbólicas são usadas para ajustar os dados, os autores em [248] explicam que não há garantia que características de não-linearidade determinadas externamente correspondem àquelas embutidas nos mecanismos que geraram os dados. A tendência é definida como uma função monotônica ajustada intrinsecamente ou uma função em que pode haver no máximo um extremo dentro de um dado período de dados. A tendência é uma das muitas propriedades locais nos dados. Portanto, necessita ser associada a uma escala de tempo para não ser confundida com ciclos locais. É demonstrado que o método EMD não somente define a tendência, mas revela algumas propriedades intrínsecas dos dados, sendo um método empírico, intuitivo, direto e adaptativo aos dados, sem requerer funções de base pré-determinadas.

Li et al [172] comparam o método EMD com uma abordagem de suavização à *priori* (*Smoothness Prior Approach* - SPA), o qual age como um filtro FIR variante no tempo passa-alta e *wavelets*, com aplicabilidade em séries artificiais de intervalo regularmente espaçadas (R-R), com quatro tipos de tendências simuladas (linear, Gauss, *cusp* e *break*) em séries temporais. Os autores citam que a presença de tendências lentas ou irregulares, em séries temporais, pode potencialmente distorcer a análise espectral, causando interpretações errôneas e que muitos métodos podem ser usados para remover tendências de séries temporais não-estacionárias regularmente espaçadas (R-R). Como resultado da comparação dos métodos, o método *wavelet* é sugerido como a melhor escolha para *detrending* de séries temporais R-R, curtas e longas.

Em [161], o algoritmo *Multi-Stage Similarity Matching* baseado no método EMD é proposto na área de mineração de dados para séries temporais financeiras, as quais geralmente são não-estacionárias e não-lineares. O algoritmo baseado em EMD

assegura que as tendências são similares, fornecendo informações úteis e relativas para os usuários, reduzindo a redundância de correspondência de similaridade.

Montesino-Pouzouls e Lendasse [193] avaliam o efeito de diferentes abordagens de *detrending* em modelos de Inteligência Computacional (CI): polinomial *detrending* de ordem 1 (*linear detrending*), o método de filtro primeira diferenciação (*first differencing*) e o método de *detrending* não-linear, baseado em *Ensemble* EMD (EEMD) [247], é um método com robustez melhorada em relação ao tradicional EMD. Segundo esses autores, o método EMD tem forte paralelismo com uma técnica amplamente usada no campo de dinâmica de fluídos, a decomposição Reynolds, fazendo uma decomposição das séries temporais, separando a média e componentes de flutuação de um sinal. Muitas abordagens de *detrending* têm sido comparadas em termos de desempenho de modelos preditivos construídos em conjuntos de dados *detrended*. Segundo os autores, o método EEMD fornece melhor desempenho que o *detrending* linear, e o método de diferenciação em alguns casos, pode ser contra-produtivo para séries temporais apresentando padrões comuns.

Moghtaderi et al [192] consideram o problema de filtrar a tendência de baixa frequência a partir das séries temporais. O problema de decompor e classificar os componentes resultantes como tendência ou flutuação (resíduo) é chamado problema de filtrar a tendência (ou de estimar a tendência), a qual é altamente dependente do contexto. É proposto um método não-paramétrico denominado *Empirical Mode Decomposition Trend Filtering*, com um procedimento para selecionar as funções de modo intrínseco e automático. São realizados testes com dados reais, considerando a remoção das tendências aditiva, usando subtração e, multiplicativa, por meio da divisão das séries originais pela tendência filtrada. Os resultados são comparados com o filtro Hodrick-Prescott e Análise do Espectro Singular - *Singular Spectrum Analysis* (SSA) [239]. Nesse caso, a tendência é definida como variando lentamente, representada por funções de modo intrínseco mais lentas, produzidas pelo método EMD. EMD *trend filtering* compartilha características comuns com SSA aplicados para o mesmo problema porque SSA realiza a decomposição em componentes oscilatórios e usa os componentes de mais baixa frequência para identificar a tendência. A tendência estimada, a partir do EMD *Trend Filtering*, é completamente dirigida aos dados, sendo descrita por um conjunto de IMFs de baixa frequência, identificando uma versão mais suave da tendência.

- *Singular Spectrum Analysis* (SSA): Elsner e Tsonis [121] citam que o termo espectro singular (*singular spectrum*) é originário da decomposição spectral (*eigenvalue*)[8] de uma matriz em seu conjunto (*spectrum*) de *eigenvalues*. Tais *eigenvalues* são os números que tornam a matriz singular. Esses autores explicam que o termo *Singular Spectrum Analysis* não é adequado, visto que a decomposição *eigenvalue* tradicional envolvendo dados multivariados, é também uma análise do espectro singular e a nomenclatura deveria ser análise de séries temporais usando o espectro singular.

  Em [121], SSA é originário da área de sistemas dinâmicos (Teoria do Caos)[9], sendo

[8]Seja A uma matriz NxN. Um número real λ é um autovalor (*eigenvalue*) real de A, se existe um vetor não nulo V de R, tal que $AV = \lambda V$. Um vetor não nulo que satisfaça a equação é chamado de autovetor (*eigenvector*) de A. Para cada autovalor λ, os autovetores associados a λ são os vetores não nulos da solução do sistema. Fonte: Instituto Gauss de Matemática. URL: http://www.igm.mat.br/aplicativos/index.php. Acesso em Junho/2014.

[9]URL:http://dbpedia.org/page/Chaos_theory. Acesso em Julho/2014

utilizado em processamento de sinais, possuindo amplo uso como uma ferramenta para análise de séries temporais, onde, diferente de outras técnicas baseadas em decomposição espectral, SSA tem como vantagem reduzir a dimensionalidade. O nome SSA foi proposto por Vautard e Ghil [239] e, segundo Alexandrov et al [59], esse método é conhecido como Abordagem *Caterpillar*.

SSA é uma abordagem para análise e predição de séries temporais. A natureza adaptativa aos dados das funções base usada pelo método se destaca dos demais métodos de análise espectral, tornando a abordagem adequada para análises de dinâmica não-linear. SSA está relacionado com a análise de componentes principais (PCA), possibilitando a remoção do ruído sem perda significante do sinal. Devido aos *eigenvectors* das séries temporais não serem assumidas como senoidais, como é o caso de funções base usando métodos *Fourier*, filtrar com SSA corresponde a um processo adaptativo aos dados. Dada a necessidade de remoção de tendência em séries temporais, se a mesma for considerada linear, o procedimento de removê-la é relativamente simples, mas esse nem sempre é o caso. O filtro baseado em SSA remove tendências não-lineares. Assim, a escolha de um filtro depende do propósito pretendido na análise [121].

Elsner e Tsonis [121] apresentam um exemplo de médias de fechamento do Índice Dow-Jones para os últimos dias úteis de cada mês, no período de Janeiro de 1952 a Junho de 1990 (Figura 2.25), onde alterações de baixa frequência dominam a variância e nenhuma simples tendência é aparente. Nesse caso, SSA pode ser usada para reconstruir o registro sem essa tendência considerada irregular.

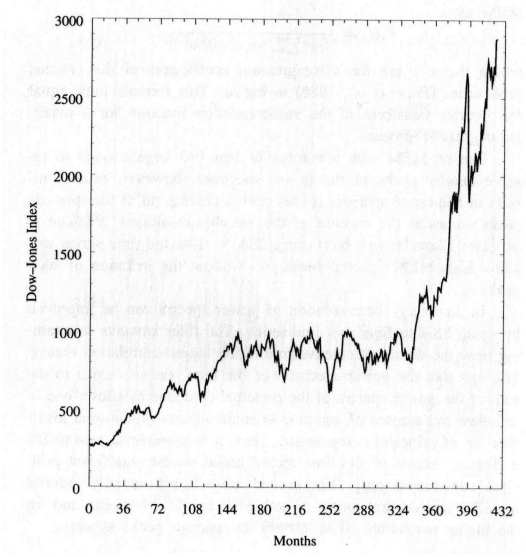


Figura 2.25: Séries temporais índice Dow-Jones (1952-1990) [121].

A Figura (2.26) mostra as séries temporais do Índice Dow-Jones reconstruída usando dois componentes principais e a Figura (2.27) apresenta, na parte superior, as séries filtradas usando SSA (filtro não-linear) e, na parte inferior, as séries filtradas linearmente pela subtração da tendência linear, estimada pelo método de mínimos quadrados ordinários (OLS). O filtro SSA remove a tendência não-linear, resultando em uma tendência ligeiramente linear e com variância em oscilações de alta frequência mais regulares de 3 a 4 anos. No caso linear, o registro apresenta inclinação zero,

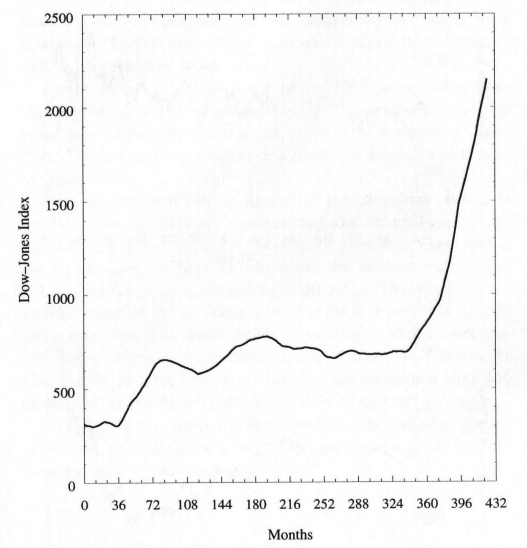


Figura 2.26: Índice Dow-Jones reconstruído por SSA [121].

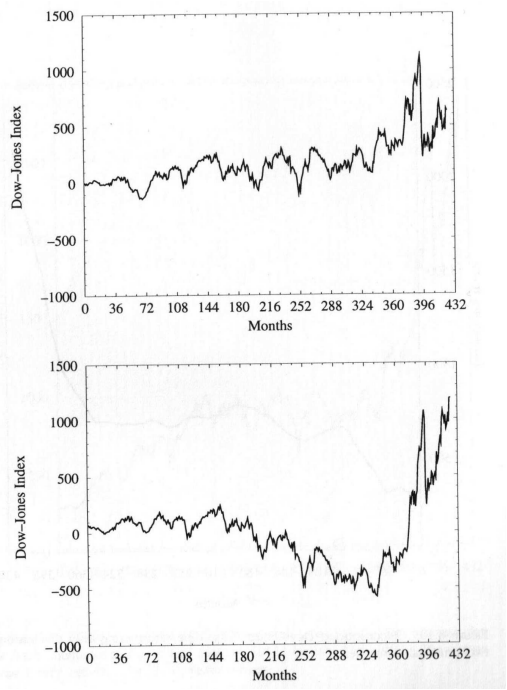


Figura 2.27: Séries temporais filtradas por SSA e linearmente filtradas por OLS [121].

apresentando grande porcentagem da variância em mudanças de baixa frequência irregulares.

Golyandina et al [136] descrevem o método SSA e técnicas relacionadas, citando que este método é extremamente útil para análise de séries temporais, embora não seja amplamente conhecido pela comunidade. SSA é um método de geometria multivariada, não sendo realmente estatístico em seus princípios metodológicos e tem se tornado uma ferramenta padrão em meteorologia e climatologia, sendo uma técnica usada em física não-linear e processamento de sinais.

Trata-se de um método essencialmente livre de modelo, sendo mais exploratório.

Objetiva a decomposição das séries temporais originais em uma soma de pequeno número de componentes interpretáveis, tais como uma tendência variando lentamente, componentes oscilatórios e um ruído (sem estrutura). A separabilidade é quem caracteriza o quão-bem diferentes os componentes podem ser separados uns dos outros. Um problema relacionado à SSA é a escolha dos parâmetros dos métodos *window length* e a maneira de agrupamento [136].

SSA é um método para identificar e extrair componentes oscilatórios, a partir das séries originais. A partir da decomposição SSA, um componente pode ser identificado como a tendência das séries temporais originais, séries oscilantes (sazonalidade) ou ruído. Séries oscilantes são séries periódicas ou quase-periódicas que podem ser ou pura ou de amplitude modulada. Ruído é qualquer série aperiódica. A tendência das séries é um componente aditivo variando lentamente nas séries com todas as oscilações removidas. SSA define tendência como um componente aditivo das séries que é não-estacionário e lentamente varia durante o período inteiro das séries que estão sendo observadas. Em linguagem de frequências, tendências geram grandes potências em baixa frequência de um periodograma [136].

Conforme Alexandrov [58], SSA é uma abordagem genérica para análise e previsão de séries temporais. SSA é similar à Análise do Componente Principal - *Principal Component Analysis* (PCA) da análise multivariada. Enquanto PCA é aplicado a uma matriz, SSA é aplicado às séries temporais, fornecendo uma representação das séries em termos de *eigenvalues* e *eigenvectors* de uma matriz feita das séries temporais. SSA pode ser usado para uma ampla variedade de tarefas [58, 193]: detecção e extração de componentes quasi-periódicos, *denoising*, extração de tendências, previsão e detecção de *change-point*.

Considerando a obtenção de algum componente aditivo específico de uma dada série temporal (como a tendência), a ideia do método SSA é embutir a série temporal em um espaço euclidiano de alta dimensão e encontrar um sub-espaço correspondente ao componente procurado e, finalmente, reconstruir o respectivo componente que corresponde ao sub-espaço. A escolha do sub-espaço é crucial em SSA, consistindo de quais componentes de decomposição do valor singular (SVD) serão usados na reconstrução. Segundo Alexandrov et al [59], existem muitos métodos no contexto de SSA para fazer a extração de tendências.

Este método consiste da decomposição das séries temporais e a reconstrução de um componente aditivo desejado. A reconstrução do componente SSA considera a série temporal inteira, onde o algoritmo usa decomposição do valor singular da matriz trajetória construída, a partir de todas as partes das séries temporais. SSA não é um método local, se comparado a um filtro linear ou métodos *wavelets*, fato que o torna robusto a *outliers*. Em relação à modelagem de tendências, SSA é uma abordagem não-paramétrica. Nesse caso, não há necessidade de uma especificação à *priori* de um modelo para as séries temporais e a tendência. Ou seja, a tendência nesse caso não é considerada nem determinística nem estocástica [59].

É proposto por Alexandrov [58] um método de extração de tendência usando Análise do Espectro Singular. SSA é um método atrativo para extração de tendências devido a dois motivos. Primeiro, não requer a especificação de um modelo para as séries temporais e a tendência e segundo, extrai tendências de séries temporais de ruído contendo oscilações de período desconhecido. O método para extração de tendências

apresentado por Alexandrov [58] herda essas propriedades de SSA, sendo descrito como fácil de usar porque requer a seleção de somente dois parâmetros, além do básico parâmetro SSA chamado *window lenght* que é usado na decomposição. Um dos parâmetros gerencia a escala da tendência extraída (limite de baixa frequência) e o outro é o valor de *threshold* específico do método.

Entre os métodos existentes de extração de tendências que podem ser usados em SSA, destacam-se uma abordagem que reconstrói a tendência, a partir dos vários primeiros componentes, a qual se aplica em muitos casos reais. Outra forma mais inteligente de selecionar a componente tendência SVD, é escolher componentes com suave e ligeiramente variantes funções ortogonais empíricas - EOFs, onde o *nth* EOF é definido como a sequência de elementos do *nth eigenvector*. Outros métodos são o uso do Coeficiente de Correlação Kendall e o método proposto por Alexandrov [58], o qual seleciona o limite de baixa frequência (que gerencia a escala da tendência extraída), ou seja, quanto mais baixo for o limite da frequência, mais lenta varia a tendência extraída, considerando um periodograma da série temporal original. Um procedimento heurístico é usado para a escolha do *threshold* de baixa frequência específico do método. Esse método é comparado ao que reconstrói a tendência usando os componentes SVD de tendência conhecidos, em um exemplo simulado de tendência exponencial e um ruído Gaussiano [58].

Alexandrov [58] cita como vantagem do método SSA, o fato de ser uma abordagem livre de modelo e, como desvantagem, a complexidade computacional para o cálculo da decomposição do valor singular. Esse custo pode ser reduzido usando computação paralela.

2.7 Considerações Sobre Métodos de Estimação de Tendências

A tendência estimada necessita ser removida a partir das séries temporais. O conhecimento semântico sobre a atividade de decomposição contribui com a observação de uma série em relação ao tempo, principalmente quanto ao uso de métodos estatísticos. Para remover o componente tendência de uma série temporal decomposta aditivamente, o método usual é subtrair a tendência estimada da série temporal original. Para uma decomposição multiplicativa, a remoção é feita por dividir as séries pelos valores da tendência [30].

Chandler e Scott [95] fazem uma discussão acerca das formas de estimação de tendências. Com modelos paramétricos, são consideradas suposições sobre os dados e, quando estas são satisfeitas, resultam em análises mais eficazes. Por outro lado, se as suposições falham, o risco de má interpretação dos dados aumenta. Nesse sentido, modelos não-paramétricos oferecem maior flexibilidade em estruturas de representação, visto que não é ajustado um modelo aos dados, pois a tendência é considerada uma função suave.

Em ambos os tipos de modelagem, a verificação das suposições ajuda a proteger contra uma má-interpretação dos dados. Modelos semi-paramétricos sugerem o uso de ambas as representações. Em qualquer Estatística, depois de estimadas e removidas as tendências, os dados se tornam corrigidos de tendências (*detrended data*) [223, 95]. Em geral, a escolha de uma das abordagens depende do contexto científico. Nesta tese, os dados gerados e *detrended* são classificados conforme o método usado para sua estimação e/ou remoção. Em particular, é relevante verificar se o analista considera o componente ruído como uma alta frequência ou como flutuações irregulares sendo parte da tendência [95].

No primeiro caso, pode ser viável a modelagem da tendência conforme um modelo

paramétrico, por exemplo, o mais elementar que é o linear, com uma estrutura de erros auto-correlacionados. Destaca-se que outros modelos de tendência podem ser considerados tais como polinomiais, trigonométricos, entre outros. Nesse caso, a tendência pode ser removida, por exemplo, por subtrair a função da tendência estimada parametricamente a partir dos dados originais, resultando em dados corrigidos de tendência por análise de regressão (*regressed detrended data*), conforme descrito em [95].

No segundo caso, é necessário o uso de métodos não-paramétricos, tais como regressão não-paramétrica local polinomial, pois irregularidades nos dados necessitam ser corrigidas para se obter uma função suave da tendência, a qual é removida, geralmente, a partir da subtração da função suave estimada das séries originais, resultando em dados suavizados corrigidos de tendência (*smoothed detrended data*) [95]. Outra situação é quando as séries temporais podem ser corrigidas por meio do uso de um filtro linear passa-baixa frequência, como o filtro de suavização de médias móveis, onde a tendência linear passa através do filtro, eliminando o ruído, gerando (*smoothed filter detrended data*) [223].

Por outro lado, tendências podem ser extraídas por um filtro passa alta frequência como *Differencing*, permitindo passar o componente de alta frequência que é o ruído, gerando *high pass filtered detrended data*.

As séries temporais também podem ser corrigidas de ruído de alta frequência usando um filtro de suavização, gerando dados filtrados (*filtered data*), onde a partir destes dados, uma estimação do componente tendência pode ser feita e extraída a partir das séries filtradas.

Com o objetivo de adicionar conhecimento semânticos às séries temporais, o próximo capítulo aborda sobre Ontologias.

CAPÍTULO 3

ONTOLOGIAS

3.1 Introdução

Ontologias [143] constituem definições formais de vocabulários que permitem definir classes e propriedades de recursos e relacionamentos entre seus membros [118]. Dentre as aplicações que fazem uso de ontologias, destacam-se: gerenciamento de conhecimento, comércio eletrônico, integração de aplicações de empresas, recuperação de informações, processamento em linguagem natural, engenharia de software, *eScience*, entre outros. Sahoo et al [220] destacam que ontologias podem ser utilizadas para geração de proveniência semântica.

No contexto da Web Semântica [73], ontologias permitem gerar e compartilhar conhecimento, contribuindo para a troca de informações entre pessoas, máquinas e/ou componentes de software, derivando fatos adicionais a partir de informações existentes, por meio de inferência realizada por raciocinadores lógicos - *reasoners* em bases de conhecimento.

Em uma ontologia, a ausência de informação não implica em falsidade, como ocorre com os sistemas de bancos de dados [224]. Com ontologias, somente uma afirmação explicitamente declarada pode ser inferida [149]. Em uma disjunção de duas classes, o *reasoner* não pode excluir a possibilidade que as classes podem ser equivalentes se não forem declaradas. Ou seja, usando ontologias, tudo que se necessita afirmar é preciso que seja declarado, caso contrário, resultados contraditórios podem ser gerados. Essa premissa se justifica porque não se pode assumir que algo existe, se não afirmado sua existência. O objetivo deste capítulo é descrever o estado da arte relacionado a ontologias, incluindo definições, componentes, classificação, linguagem de consulta e desenvolvimento.

3.2 Definições

Na literatura existem várias definições para Ontologia. Falbo et al [124] destacam definições usadas em Computação. Por um lado, têm-se a comunidade de Modelagem Conceitual e áreas correlatas e, do outro, as comunidades de Inteligência Artificial, Engenharia de Software e Web Semântica.

Em Modelagem Conceitual, Modelagem Organizacional, entre outras áreas relacionadas, ontologias têm sido definidas conforme sua origem filosófica, constituindo um sistema de categorias formais e filosoficamente bem fundamentado.

Em Inteligência Artificial e áreas correlatas, ontologias representam o conhecimento, segundo dois aspectos. Primeiro, como um artefato concreto de engenharia, não fundamentado, sendo projetado para um propósito específico; e, segundo, como a representação de um domínio particular, formalizado em alguma linguagem de representação de conhecimento, como *Resource Description Framework* - RDF e *Web Ontology Language* - OWL.

No contexto da Inteligência Artificial é proposta a definição por Gruber [141], sendo amplamente utilizada como uma especificação explícita de uma conceitualização. Borst [76] ampliou esta definição com um elemento colaborativo, redefinindo como uma especificação formal e explícita de uma conceitualização compartilhada. É caracterizada como um meio de representação do conhecimento, estruturando informações do domínio em uma maneira organizada e gerando semântica nos dados.

Kiryakov [165] define os elementos de uma ontologia como um conjunto de classes que representam conceitos no domínio de interesse; conjunto de relacionamentos entre conceitos do domínio; conjunto de instâncias de classes (derivadas e concretas) e conjunto de axiomas[1] do domínio, usados para modelar restrições e regras para instâncias. Adicionando-se à ontologia um conjunto de instâncias individuais de classes têm-se uma base de conhecimento.

3.3 Componentes

Uma ontologia em linguagem OWL consiste de classes, indivíduos e propriedades, permitindo que esses componentes sejam anotados com informações, tais como comentários, data da criação, autor, entre outras. Na sequência esses componentes são explicados para o desenvolvimento de ontologias OWL, conforme [152, 149, 93].

3.3.1 Conceitos e Instâncias

Conceitos constituem as classes, sendo os principais componentes de uma ontologia, as quais são interpretadas como conjuntos que contêm indivíduos. Um conceito é um objeto abstrato que define características comuns de um grupo de objetos concretos (instâncias ou indivíduos). Um conceito pode ser definido como a união de outros conceitos. OWL não usa Suposição de Nomes Únicos - *Unique Name Assumption* (UNA), ou seja, deve ser explicitamente afirmado que indivíduos são os mesmos ou são diferentes. OWL permite descrever um domínio em termos de:

- Indivíduos ou Instâncias - objetos particulares do domínio.
- Classes - coleções de objetos que compartilham características semelhantes.
- Propriedades - relacionamentos binários entre indivíduos.

Além destes, inclui também uma coleção de axiomas descrevendo como essas classes, indivíduos e propriedades deveriam ser interpretados. OWL tem um número de operadores que permitem descrever as classes e suas características, tais como operadores booleanos - *and* (e), *or* (ou) e *not* (negação); quantificação sobre propriedades/relacionamentos (universal, existencial) e uma semântica clara e não-ambígua para operadores e expressões de classes.

O uso de ontologias varia a partir de uma classificação base, usando conceitos primitivos para inferência com conceitos definidos. Uma classe que tem somente condições necessárias é definida como Classe Primitiva e uma classe que tem, no mínimo, um conjunto de condições necessárias e suficientes é conhecido como Classe Definida, onde qualquer indivíduo que satisfaça essa definição pertencerá a esta classe. Um *reasoner* somente classifica automaticamente instâncias em classes definidas.

Em uma ontologia formal, conceitos podem ser primitivos ou definidos, fazendo uso de condições necessárias e suficientes e/ou restrições por condições necessárias, expressas como restrições em propriedades.

Classes podem ser definidas como disjuntas - *disjoint*, onde um mesmo indivíduo não pode ser instância de duas classes disjuntas. *Reasoners* podem verificar a consistência

[1] axioma é uma afirmação que pode ser vista como verdade sem necessidade de provas.

dessas classes, como quando se declara uma classe como subclasse de duas classes distintas. Nesse caso, após a inferência em uma ferramenta de desenvolvimento, essas classes são identificadas para serem corrigidas.

Classes podem ser definidas pela listagem dos seus indivíduos membros, conhecidas como Classes Enumeradas.

3.3.2 Propriedades

Uma propriedade representa um relacionamento binário entre dois conceitos ou uniões de conceitos. O domínio de uma propriedade é o conceito para o qual a propriedade pode ser aplicada e o intervalo de uma propriedade é o conceito onde a propriedade obtém valor.

Em OWL, propriedades podem ser de Objeto, de Tipos de Dados - *Datatypes* ou de Anotações. Propriedades de Objetos são relacionamentos entre dois indivíduos. Propriedades *Datatype* relacionam um indivíduo para um valor de um tipo de dado definido em um *Schema* XML ou um literal RDF, descrevendo relacionamentos entre indivíduos e seus valores de dados. Propriedades de Anotações adicionam informações para classes, indivíduos e/ou Propriedades de Objeto ou de Tipos de Dados.

Propriedades de Objetos apresentam características, podendo ser Inversas, Funcionais, Funcionais Inversas, Transitivas, Simétricas, Assimétricas, Reflexivas e Irreflexivas [149].

Algumas vezes a existência de um relacionamento em uma direção implica que outro relacionamento existe em direção oposta ou inversa, caracterizando as Propridades Inversas, tais como *hasPart* e *isPartOf*. Propriedades de Objeto podem ser limitadas a ter um único valor, sendo Funcionais, por exemplo a data de nascimento ou a mãe biológica de um indivíduo. Um exemplo de propriedade Funcional Inversa é *hasEmailAddress*, indicando que nenhum de dois indivíduos pode ter o mesmo valor para aquela propriedade, mas um indivíduo pode ter mais de um valor único.

Uma propriedade de Objeto também pode ser Transitiva, como *contains*. Nesse caso, se a propriedade p descreve um indivíduo a relacionado a um indivíduo b e b é relacionado a c, é possível inferir que o indivíduo a é relacionado ao indivíduo c, via propriedade p.

Uma propriedade Simétrica é quando a propriedade descreve um indivíduo a para um indivíduo b, então o indivíduo b é também relacionado ao indivíduo a via propriedade p, por exemplo *equals*. A propriedade Assimétrica nunca existe como um relacionamento bidirecional, onde nenhum de dois indivíduos a e b podem ser relacionados (a p b) e (b p a) por uma propriedade Assimétrica p, por exemplo *isGreatherThan*.

Uma propriedade p é dita ser Reflexiva quando relaciona um indivíduo a para si mesmo, por exemplo a propriedade *knows*. Uma propriedade Irreflexiva nunca relaciona um indivíduo a si mesmo, por exemplo *hasMother*. Propriedades podem ser Reflexivas, Irreflexivas ou nenhuma destas.

A Tabela 3.1 apresenta a descrição dessas propriedades onde A, B e C são classes, x e y representam literais ou indivíduos e p representa uma propriedade do tipo correspondente.

Uma restrição descreve classes de indivíduos baseado nos relacionamentos que participam. Em OWL existem as seguintes restrições: Restrições de Quantificador, Restrições de Cardinalidade e Restrições *hasValue*.

Restrições de Quantificador podem ser categorizadas em Restrições Existencial e Universal. No primeiro caso descreve classes de indivíduos, que participam de no mínimo um relacionamento junto com uma propriedade especificada a indivíduos que são membros de uma classe especificada. A palavra-chave utilizada é *at least one* (algum). No segundo caso, Restrições Universais descrevem classes de indivíduos que, para uma dada proprieda-

Tabela 3.1: Classes de propriedades OWL [149].

Propriedade	Definição
owl:InverseProperty	(Entity1 p1 Entity2) implica em (Entity2 p2 Entity1)
owl:FunctionalProperty	(A p x) e (C p y) implica em x = y
owl:InverseFunctionalProperty	(A p B) e (C p B) implica em A = C)
owl:TransitiveProperty	(A p B) e (B p C) implica na declaração (A p C)
owl:SimmetricProperty	(A p B implica na declaração B p A)
owl:AsimmetricProperty	(A p B implica que não há declaração B p A)
owl:ReflexiveProperty	(implica a declaração (A p A), para todo A)
owl:IrreflexiveProperty	(implica que não há declaração (A p A), para todo A)

de, têm relacionamentos com esta propriedade, somente junto a indivíduos que são membros de uma classe especificada. A palavra-chave utilizada é *only* (somente).

As Restrições de Cardinalidade especificam o número exato de relacionamentos que um indivíduo participa de relacionamentos com outros indivíduos ou valores de tipos de dados, podendo ser mínima - *at least*, máxima - *at most* ou exatamente - *exactly*. Restrições de Cardinalidade Qualificada são mais específicas que as de Cardinalidade, onde se declara a classe de objetos dentro da restrição.

Restrições de Valor são denotadas pelo símbolo $\ni$, descrevendo o conjunto de indivíduos que tem, no mínimo, um relacionamento junto com uma propriedade especificada para um indivíduo específico.

Devido a *Open World Assumption*, até que se afirme explicitamente que indivíduos de uma classe não podem ter outros relacionamentos com outras classes, o *reasoner* pode inferir o contrário. Para essa afirmação são utilizados axiomas de fechamento nas propriedades. Um axioma de fechamento em uma propriedade consiste de uma restrição universal que age junto com a propriedade para dizer que somente pode ser formada como especificada. A restrição apresenta a união das Restrições Existenciais que podem ocorrer para a propriedade.

3.3.3 Conceitos e Propriedades Subordinadas

Conceitos e propriedades subordinadas são organizados em uma hierarquia de relacionamento de subordinação de posição, sendo um tipo de relacionamento *is a*, ou seja, classes ou propriedades filhas são mais específicas do que suas classes ou propriedades pais.

- *Subordinação de Conceitos*: Conceitos podem ser organizados em uma hierarquia superclasse-subclasse, conhecida como taxonomia. Subclasses especializam, ou seja, são subordinadas, pelas suas superclasses. Se A e B são dois conceitos, diz-se que B é subordinado por A se e somente se todas as instâncias de B são instâncias de A. O subordinador universal é denominado *Thing* ou *TOP*, encontrado no topo de uma hierarquia de subordinação.

- *Subordinação de Propriedades*: A notação (B $\rightarrow$ A significa que "B é subordinado por A"), pois se A e B são duas propriedades, B é subordinado por A se e somente se o domínio (B) é subordinado pelo domínio (A), assim como o intervalo (B) é subordinado pelo intervalo (A).

3.3.4 Regras

Horrocks et al [154] estendem axiomas OWL para permitir axiomas de regras (*Semantic Web Rule Language* - SWRL), na forma *axioma::=regra*. De modo informal, uma regra tem a forma antecedente → consequente, uma implicação entre um antecedente (corpo) e consequente (cabeçalho).

Informalmente, uma regra significa "se o antecedente é verdade, então o consequente deve ser verdade". O antecedente e consequente de uma regra consistem de zero ou mais átomos, podendo estar nas formas C(x), P(x,y), *sameAs*(x,y) ou *differentFrom*(x,y), onde C é uma descrição OWL, P é uma propriedade OWL, x e y são variáveis, indivíduos ou valores de dados. Múltiplos átomos em um antecedente são tratados como uma conjunção e múltiplos átomos em um consequente são tratados como consequências separadas.

A Ferramenta Protégé 4.1 [166] permite trabalhar com regras a partir de *View Rules*. O *reasoner* Pellet [227] dá suporte a inferência com regras SWRL, as quais interpretam SWRL usando a noção de regras seguras - *DL-Safe Rules*, onde estas serão aplicadas, somente, para indivíduos nomeados na ontologia.

3.4 Classificação de Ontologias

Ontologias podem ser classificadas segundo:

- Formalidade [236]: *Altamente informal*, usando linguagem natural; *Semi-informal*, usando linguagem natural de forma estruturada; *Semi-formal*, usando linguagens artificiais formalmente definidas, como UML; e *Rigorosamente formal*, apresentando semântica formal, teoremas e provas, tal como Lógica de Primeira Ordem - *First Order Logic* (FOL).

- Estrutura de conceitualização [238]: *Terminológica*, especificando conceitos de determinado domínio; *Informacional*, especificada conforme um esquema de base de dados; e *Representação de Conhecimento*, especificando a conceitualização do conhecimento para uso particular.

- Assunto de conceitualização [238]: *Aplicação*, definindo conceitos para modelagem do conhecimento em determinado domínio de aplicação; *Domínio*, definindo conceitos e relações em domínio específico; *Genéricas*, definindo um domínio geral; e *Representação do Conhecimento*, formalizando o conhecimento sobre determinado paradigma.

- Nível de generalidade [143]: *Alto Nível*, descrevendo conceitos genéricos; *Domínio*, fornecendo vocabulário sobre um domínio genérico; *Tarefa*, definindo conceitos relacionados a uma tarefa específica; e *Aplicação*, descrevendo conceitos pertencentes a um domínio e a uma tarefa.

- Riqueza de sua estrutura interna [170]: *Vocabulários controlados*, listando termos; *Glossários*, listando termos e seus significados; *Tesauros*, usando semântica adicional entre os termos, inclusive sinônimos; *Hierarquia informal*, com hierarquia de termos não correspondendo a uma classe restrita; *Hierarquia formal*, com hierarquia de termos correspondendo a uma classe restrita; *Frames*, incluindo classes e propriedades; *Restrição de valores*, com uso de restrições em valores associados às propriedades;

e *Restrição lógica geral*, usando linguagem bem expressiva, permitindo restrições entre termos, por meio de Lógica de Primeira Ordem.

- Tipo de modelagem da ontologia [93]: *Lightweight*, modelando taxonomias; e *Heavyweight*, modelando restrições semânticas sobre o domínio, utilizando axiomas e restrições para ontologias *Lightweight*.

3.5 Linguagem de Consulta SPARQL

A Linguagem SPARQL[2], atualmente, encontra-se na Versão 1.1, sendo acrônimo para *SPARQL Protocol and RDF Query Language* [213]. É uma recomendação do Consórcio W3C, desde 2008, possuindo formato como *Structured Query Language* - SQL para consultas a dados RDF. A Ferramenta Protégé [166] Versão 3.4 possui SPARQL integrada. A Versão 4.1 de Protégé possui o *Plugin OWL2 Query Tab* [2] que permite a geração de grafos e consultas em SPARQL. O *Plugin DL Query* permite especificar consultas de forma mais simples, na sintaxe de Manchester-OWL.

A Linguagem SPARQL utiliza um padrão de triplas RDF (*Subject*, *Predicate*, *Objects*), onde uma tripla é a estrutura de dados básica de RDF [118]. *Subject* representa um *Universal Resource Identifier* - URI, identificando um recurso de forma única. *Predicate* representa a propriedade que identifica o recurso, por meio de uma URI. *Object* representa uma tripla, constituindo um valor de atributo, o qual pode ser um literal, uma URI ou um nodo vazio. Padrões de grafos são conjuntos de padrões de triplas, onde consultas são lidas a partir dos mesmos. Abaixo tem-se um exemplo de consulta em SPARQL.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?person, ?age
FROM <http://example.org/data.rdf>
WHERE  ?person a foaf:Person ; foaf:age ?age .
ORDER BY ?age DESC LIMIT 5
```

Este exemplo utiliza o Vocabulário *Friend Of a Friend* - prefixo *foaf*, consultando pessoas (?person) e idades (?age), ordenando de forma decrescente pela idade, com limite de cinco elementos.

3.6 Desenvolvimento de Ontologias

Esta seção aborda os princípios para o desenvolvimento de uma ontologia, assim como metodologias existentes, mediação de ontologias, implementação e limitações de desenvolvimento.

3.6.1 Princípios

Gruber [141] destaca princípios a serem seguidos no desenvolvimento de uma ontologia:

- *Clareza*: definições em uma ontologia devem ser claras e bem documentadas.

[2]SPARQL pronuncia-se *"sparkle"*

- *Coerência*: no caso de ontologias *heavyweight*, axiomas precisam ser consistentes, para que inferências estejam em consoância com o domínio.
- *Extensibilidade*: definições devem ser extensíveis.
- *Codificação mínima*: conceitos especificados não devem ser dependentes de linguagens específicas.
- *Compromisso ontológico mínimo*: definição terminológica deve ser suficiente para permitir compartilhamento de informações.

3.6.2 Metodologias

Existem várias metodologias para o desenvolvimento de ontologias, destacando-se: *Enterprise Ontology* [237]; *TOVE - Toronto Virtual Enterprise* [142]; *METHONTOLOGY* [127]; *On-To-Knowledge* [231]; *REFSENO* [235]; *Ontology Development 101* [204] e *NeOn Methodology* [233].

Não existe um consenso geral da comunidade sobre uma metodologia que seja adotada como padrão [208], visto que cada uma apresenta especificidades da Engenharia de Ontologias [106]. Fernández-Lópes e Gómes-Pérez [126] recomendam que pode ser adotado o uso de uma combinação de metodologias. A *METHONTOLOGY* é considerada a metodologia mais madura, recomendada pela *Foundation for Intelligent Physical Agents* - FIPA. Na sequência são abordados os passos metodológicos inerentes às metodologias.

- *Enterprise Ontology*: identificar a proposta da ontologia; construir a ontologia capturando, codificando e integrando conhecimento a partir de ontologias existentes; avaliação; e documentação da ontologia.
- *TOVE (Toronto Virtual Enterprise)*: capturar cenários motivacionais; formulação de questões de competência informais; identificação da terminologia formal; elaboração de questões em FOL; especificação de axiomas e avaliação da ontologia.
- *METHONTOLOGY*: especificar requisitos; conceitualização do domínio do conhecimento; formalização do modelo conceitual em uma linguagem formal; implementação do modelo formal e manutenção de ontologias implementadas.
- *On-To-Knowledge*: levantar e especificar requisitos e questões de competência, reuso de ontologias e desenvolvimento de uma versão preliminar da ontologia; refinamento: onde é construída uma ontologia madura e orientada à aplicação; avaliação: envolve a verificação dos requisitos e questões de competência, assim como testes da ontologia no ambiente da aplicação; e manutenção.
- *REFSENO (Representation Formalism for Software Engineering Ontologies)*: esta metodologia é uma adaptação do METHONTOLOGY, o desenvolvimento é feito a partir de um guia representacional que utiliza tabelas e diagramas, envolvendo: planejamento; especificação dos requisitos; conceitualização e implementação, onde é feita a representação e armazenamento do passo anterior por meio de ferramentas computacionais.

- *Ontology Development 101*: determinar o domínio e escopo da ontologia; identificar a partir do domínio um conjunto de questões de competência que a ontologia deve responder; considerar reuso; enumerar itens importantes; definir classes e hierarquias de classes; definir propriedades e restrições; e definir instâncias.

- *NeOn Methodology*: essa metodologia é centrada na modularização de ontologias. Descreve nove cenários flexíveis relacionados ao desenvolvimento colaborativo, dando especial ênfase no reuso e na re-engenharia de recursos de conhecimento, ontológicos e não-ontológicos.

3.6.3 Mediação de Ontologia

A Mediação de Ontologia - *Ontology Mediation* torna explícito os relacionamentos entre diferentes ontologias, contribuindo para uma aplicação usar diferentes ontologias. Este processo reconcilia diferenças entre ontologias heterogêneas a fim de alcançar interoperabilidade entre fontes de dados anotadas com aplicações que utilizam essas ontologias. Isto inclui a descoberta e especificação de mapeamentos de ontologias, como o uso desses mapeamentos para certas tarefas, tais como reescrita de consultas e transformação de instâncias [113, 123] .

A Mediação de Ontologias é um campo amplo de pesquisa, o qual determina diferenças entre ontologias a fim de permitir reuso de dados através de aplicações heterogêneas. A mediação de ontologias pode ser subdividida em três áreas [112]:

- Mapeamento de Ontologias: o mapeamento de ontologias é utilizado para interoperabilidade, sendo o processo de encontrar correspondências entre conceitos de duas ontologias [114]. Trata-se de uma função associando termos e expressões de uma ontologia-fonte a termos e expressões de uma ontologia-alvo [1] (Figura 3.1).

Figura 3.1: Mapeamento de ontologias [113].

- Alinhamento de Ontologias: o alinhamento de ontologias investiga a descoberta (semi-)automática de correspondências entre ontologias, preservando as ontologias originais [205]. Bruijn [113] afirma que é o processo que traz as ontologias a um acordo comum. As ontologias são mantidas separadas, mas pelo menos uma das ontologias originais é adaptada, tal que a conceitualização e o vocabulário correspondendo a partes semanticamente sobrepostas das ontologias. No entanto, as ontologias podem descrever diferentes partes do domínio, em diferentes níveis de detalhes.

  O alinhamento de uma ontologia M é uma especificação declarativa da sobreposição semântica entre duas ontologias A e B, o qual pode ser unidirecional ou bidirecional. No primeiro caso especifica-se como expressar termos em A usando termos a partir de B de modo não facilmente inversível. No segundo caso o mapeamento trabalha com ambas as formas, onde um termo em A é expresso usando termos de B e no sentido inverso [113] (Figura 3.2).

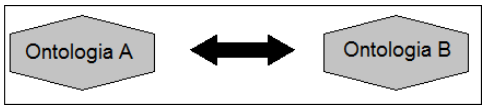


Figura 3.2: Alinhamento de ontologias [113].

- Junção de Ontologias: relacionada com a criação de uma nova ontologia, baseada em um número de ontologias-fonte. Neste caso, a nova ontologia unifica e substitui as ontologias originais, requerendo considerável adaptação e extensão (Figura 3.3).

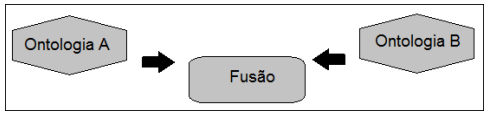


Figura 3.3: Junção de ontologias [113].

Esta definição não diz como as ontologias integradas se relacionam com as ontologias originais. As abordagens mais utilizadas são a união e a interseção. Na operação de união, a ontologia integrada é a união de todas as entidades de ambas ontologias-fonte, onde diferenças na representação de conceitos similares têm sido resolvidas. Com a operação interseção, as ontologias integradas consistem somente de partes das ontologias-fonte que foram sobrepostas, seguindo o conceito do operador de interseção da álgebra relacional [113].

3.6.4 Implementação

A implementação de uma ontologia envolve escolhas para seu desenvolvimento, dada a diversidade de ferramentas com diferentes capacidades e limitações. A seguir são descritas linguagens de representação, *reasoners* e editores de ontologia.

- *A linguagem de representação*

  A formalização para conceitos definidos necessita de linguagens que possibilitem seu desenvolvimento. Lógicas Descritivas (LDs) [67] são consideradas um meio adequado e maduro de representar ontologias. A Linguagem OWL é uma recomendação do Consórcio W3C desde 2004, sendo utilizada para representar o significado de termos e seus relacionamentos em vocabulários. É uma linguagem intuitiva para humanos e processável por máquinas. Seu uso é mais fácil para expressar significados e semântica, comparada à *eXtensible Markup Language* - XML, *Resource Description Framework* - RDF e *Resource Description Framework Schema* - RDFS. A Linguagem OWL foi recentemente padronizada como uma linguagem de ontologia orientada a Web, a qual é baseada em LD, sendo a linguagem mais utilizada para descrever ontologias. Esta linguagem apresenta três sub-linguagens OWL, classificadas conforme sua expressividade [152]: OWL-*Lite*, OWL-DL e OWL-*Full*.

  OWL-*Lite* é a menos expressiva e sintaticamente mais simples, sendo usada em situações onde somente uma hierarquia de classes e simples restrições são necessárias.

  OWL-*Full* é a mais expressiva das sub-linguagens. Seu uso envolve situações onde alta expressividade é mais importante do que decidibilidade. Uma característica importante é que não é possível inferências automatizadas em ontologias OWL-Full.

OWL-DL é muito mais expressiva que OWL-*Lite*, sendo ambas baseadas em LD, um fragmento decidível[3] de Lógica de Primeira Ordem, contribuindo para *reasoning* automático. É possível computar, automaticamente, a hierarquia de classificação (relacionamento de subordinação) e verificar inconsistências em ontologias consoantes com OWL-DL. OWL-DL pode ser considerada uma extensão de OWL-*Lite*, da mesma forma, OWL-*Full* uma extensão de OWL-DL.

- *Inferências por um Reasoner*

  Inferências são feitas por um *reasoner*, uma ferramenta que infere informações de forma implícita envolvendo consistência, subordinação, equivalência e instanciação. Existem vários *reasoners* disponíveis, destacando-se: Racer, FaCT, HermiT e Pellet.

  A Tabela 3.2[4] mostra *reasoners* que são gratuitos e otimizados para tarefas de inferências considerando a Lógica de Descrição $\mathcal{SROIQ}$ [153], incluindo sua denominação, licença de uso, linguagem de desenvolvimento, algoritmo que implementa e a respectiva API para uso por uma linguagem de programação.

Tabela 3.2: *Reasoners* in $\mathcal{SROIQ}$

Reasoner	Licença	Linguagem	Algoritmo	API
FaCT++	GPL/LGPL	C++	*tableau-based decision procedure* (Tboxes/Aboxes)	OWL-API, lisp-API, DIG interface.
HermiT	LGPL	Java	*hypertableau-based decision procedure*	OWL API 3.0
Pellet	open-source	Java	*tableau-based decision procedure* (Tboxes/Aboxes)	OWL-API, DIG e Jena interface.

- *Editores de Ontologias*

  Existem várias ferramentas para implementação de ontologias. Para escolha de qual ferramenta utilizar, é preciso conhecer suas especificações e propósitos, *plugins* existentes, entre outros. A seguir são descritas as ferramentas Protégé [166] e SWOOP [162].

  – *Protégé*

    Esta ferramenta foi desenvolvida pela Universidade de Stanford. Trata-se de uma ferramenta desenvolvida em Java, bem documentada, utilizada amplamente, apresenta uma interface gráfica, ambiente *plug-and-play*, exporta ontologias para vários formatos tais como RDF, RDFS, OWL, XMLSchema, Sintaxe Manchester OWL, entre outros. Atualmente a versão disponível é 4.1 [152].

  – *SWOOP*

    Esta ferramenta hipermídia possui código aberto, produzido pelo laboratório MIND, da Universidade de Maryland, College Park. Permite a criação, a edição e o *debugging* de Ontologias OWL. Apresenta como principais características ser um *browser* de edição de ontologias na Web, baseado no Paradigma *Model-View-Controller* - MVC e desenvolvida em Java. Apresenta arquitetura baseada em *Plugins*. O *layout* se assemelha a um *Website* baseado em *frames*, visualizado

[3] ou seja, quando computações baseadas em lógica terminarão em tempo finito.
[4] URL:http://www.cs.man.ac.uk/šattler/reasoners.html Acesso em jun/2014.

através de um Web *browser*. Essa ferramenta leva vantagem de executar em um *browser* padrão Web. Considera URIs essenciais para o entendimento e construção de Ontologias OWL.

3.6.5 Limitações

Algumas limitações podem ser identificadas e tratadas no desenvolvimento de uma ontologia, tais como [93]:

- *Cardinalidade e cardinalidade qualificada*

  Cardinalidade descreve uma restrição quanto ao número de vezes que uma propriedade tem um conceito como seu domínio. Cardinalidade qualificada também possibilita declarar o intervalo de uma propriedade, especificando o tipo da classe relacionada. Cardinalidade é permitida, assim como a cardinalidade qualificada. Porém o uso desta última é desaconselhado, pois é *CPU-heavy*, devendo ser substituída por restrições existenciais sempre que possível.

- *Intervalos e Enumeração*

  Intervalos e enumeração podem degradar o desempenho da ontologia e devem ser usados com cautela.

- *Complexidade deve ser adequada*

  Se a estrutura da ontologia é difícil de inferir pelo *reasoner*, como no caso onde existem muitas restrições *CPU-heavy*, tais como cardinalidade qualificada, enumerações, entre outras, mesmo no caso de a ontologia ser bem elaborada, a exploração em aplicações será comprometida, visto que o tempo de inferência será muito longo, onde a mesma deve ser revista.

- *Definições devem ser adequadas*

  Definições e restrições devem se encaixar no uso da ontologia. Nesse caso classes não utilizadas devem ser descartadas. Outra observação importante é que uma ontologia usável não é uma descrição universal, pois é impossível se ter uma representação perfeita e ideal e mesmo se isso fosse possível, a complexidade será tão alta que a estrutura se torna impossível para usar e manter.

- *Tamanho deve ser gerenciável*

  Uma ontologia que é detalhada excessivamente ou que aborda um campo muito amplo pode tornar-se ilegível, difícil de manter e produzirá tempos de inferência inaceitáveis.

Guarino [143] destaca que, idealmente, ontologias de domínio deveriam ser construídas com base em ontologias de fundamentação [149], pois estas são teoricamente bem fundamentadas, constituindo-se de sistemas de categorias independentes de domínio, descrevendo conceitos gerais, e podendo ser utilizadas para melhorar a qualidade de modelos conceituais, inclusive de ontologias de domínio.

3.7 Desenvolvimento de Aplicações

O desenvolvimento de aplicações que explorem ontologias OWL é feito por meio de uma *Application Program Interface* - API. Na literatura destacam-se as seguintes APIs: Jena *Framework* [16], Protégé-OWL API [27] e OWL API [25].

- Jena *framework*

  Este *framework* foi desenvolvido na Incubadora Apache e a partir de Abril de 2012 foi aprovado como um projeto de alto nível Apache, desenvolvido para aplicações da Web Semântica. Apresenta como principais características ser *open source*, apresenta maturidade, oferece compatibilidade com a maioria das APIs para RDFS/-OWL, um dos motivos de ser amplamente utilizado. Fornece uma coleção de ferramentas e bibliotecas Java, incluindo:

  - uma API para leitura, processamento e escrita de dados RDF em XML, N-*triples* e formatos Turtle;
  - uma API para tratar ontologias OWL e RDFS;
  - um motor de inferência, baseado em regras, para *reasoning* com fontes de dados RDF e OWL;
  - um grande número de triplas RDF, armazenadas em disco, de modo eficiente;
  - um motor de consulta com a mais recente especificação SPARQL;
  - servidores permitindo que dados RDF sejam publicados para outras aplicações, usando uma variedade de protocolos, incluindo SPARQL.

  Uma limitação deste *framework* é sua generalidade para RDF/RDFS, não fornecendo primitivas específicas para aplicações OWL.

- Protégé OWL API

  Esta API fornece funções para trabalhar com aplicações OWL, de modo mais fácil e mais simples do que a API Jena em Java. Foi desenvolvida em consoância com a API *core* do Editor Protégé. Esta API pode ser usada diretamente, por aplicações externas, para acessar bases de conhecimento Protégé.

  Protégé-OWL API é uma biblioteca Java *Open Source* para OWL e RDF(S). Esta API fornece classes e métodos para carregar e salvar arquivos OWL, consultar e manipular modelos de dados OWL, assim como permite *reasoning*. É otimizada para a implementação de interfaces gráficas e projetada para uso em dois contextos principais:

  - Desenvolvimento de componentes que são executados dentro da interface de usuário do Editor Protégé-OWL;
  - Desenvolvimento de aplicações *stand-alone* como aplicações *Swing*, *Servlets* ou *plugins* Eclipse.

- OWL API

  Esta API está em desenvolvimento, a qual trata exclusivamente OWL, mais precisamente versões OWL2. Enfatiza desempenho com uma implementação eficiente

de representações, oferecendo melhor integração de *reasoners*. É *Open Source*, disponível sob a Licença LGPL - *GNU Lesser General Public License* ou *Apache Licenses*, sendo desenvolvida em Java. O gerenciador *OWLOntologyManager* é utilizado para carregar e criar ontologias. Esta API inclui os seguintes componentes:

- Uma API para OWL2 e uma implementação eficiente de referência *in-memory*.
- *Parser* e *writer* RDF/XML, OWL/XML, Sintaxe Funcional OWL, Turtle, KRSS e para formato de arquivo OBO Flat.
- Interfaces de *reasoner* como FaCT++, HermiT, Pellet e Racer.

Sua versão original para OWL 1.0 foi desenvolvida como parte do projeto *WonderWeb*. A Versão 2.0 foi desenvolvida como parte dos projetos CO-ODE e TONES. A Versão 3 é desenvolvida na Universidade de Manchester. Esta API foi projetada para suporte à manipulação de ontologias no nível de axiomas terminológicos, mas grande quantidade de instâncias de dados pode causar problemas. Foi projetada para dar suporte OWL e não RDF, assim como não é projetada para produção em nível industrial.

Para criação dos elementos das ontologias, existem vocabulários e taxonomias que podem ser utilizados como base para seu desenvolvimento, tais como *Provenance Vocabulary Core Ontology* [28], vocabulário de proveniência que fornece classes e propriedades para descrever proveniência de dados Web; *Dublin Core Metadata Initiative* - DCMI [9], padrões de metadados para várias aplicações; *Web of Trust* [45], estabelecendo a autenticidade entre uma chave pública e seu proprietário; *Friend of a Friend* - FOAF [11], para compartilhamento de informações sobre amigos na Web; *The Statistical Core Vocabulary* (SCOVO) [37], apresentando três conceitos principais: *Dataset*, representando o *Container* de algum dado, como uma tabela, um *Data Item*, representando um simples dado, como a célula de uma tabela e *Dimension*, representando algum tipo de unidade de um simples dado, como um período de tempo; GeoRSS [12], representando informações geoespaciais, as quais podem ser importadas e/ou reutilizadas no desenvolvimento de ontologias, entre outros.

Como forma de interoperabilidade quanto à proveniência, modelos de proveniência fundamentados em ontologias apresentam padronização e contribuem para interoperabilidade, os quais são abordados ao final do próximo capítulo.

CAPÍTULO 4

PROVENIÊNCIA

4.1 Introdução

Proveniência significa origem ou fonte de algo [194]. Possui aplicabilidade em diversas áreas, contribuindo, por exemplo, para determinar a autencidade de uma obra de arte. Em Ciência da Computação, a proveniência descreve a fonte e derivação dos dados.

Muitos *datasets* científicos são o resultado de análises complexas ou simulações, onde a proveniência é essencial para assegurar a confiança nos dados. Buneman e Tan [89] afirmam que manter um registro completo de como a computação foi feita contribui para: i. assegurar repetibilidade; ii. catalogar os resultados; iii. evitar duplicação de esforços e iv. recuperar a fonte dos dados a partir de sua saída. Também o conhecimento sobre a origem de um elemento de dados é essencial na avaliação da qualidade de um banco de dados.

Informações de proveniência provam a corretude dos dados resultantes, determinam a qualidade e a confiança nos resultados, sendo consideradas por Tan [234] tão importantes quanto os resultados em si. Essas informações podem ser utilizadas para assegurar a qualidade e autenticidade dos dados gerados, permitir reprodutibilidade e reuso, assim como gerar semântica nos dados, para fins de interoperabilidade.

A literatura ampla sobre proveniência demonstra que trata-se de um problema multidimensional [194], onde anotações podem ser propagadas de formas diferentes, utilizando operadores complexos, como em consultas aninhadas. A proveniência pode ser representada em múltiplas formas, como um conjunto de tuplas. Pode apresentar diferentes propriedades, como um conjunto mínimo ou não. A implementação pode ser de forma *preguiçosa*, gerada sob demanda, por meio de consultas requisitadas, ou ainda, de forma *ansiosa*, propagando informações em tempo de execução.

Islam [157] descreve o ciclo de vida de proveniência, onde aplicações de proveniência (Figura 4.1) criam documentação de execução, descrevendo o que ocorreu em tempo de execução.

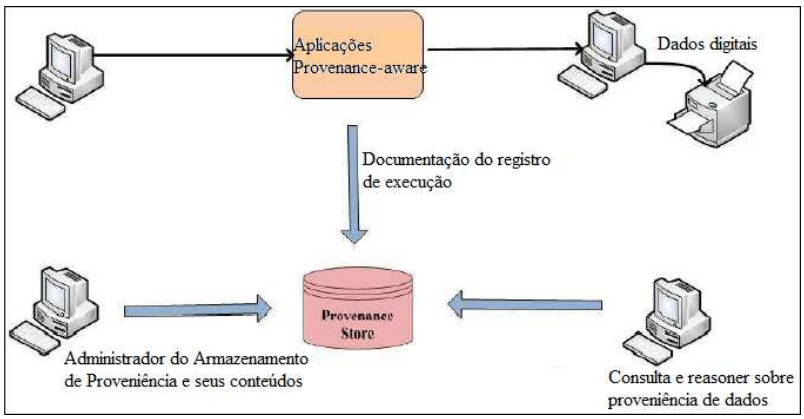


Figura 4.1: Ciclo de vida de proveniência [157] (tradução).

Um repositório persistente e seguro é usado para armazenar essa documentação dos

processos. Seus conteúdos precisam ser gerenciados ou mantidos e podem ser consultados para recuperar e analisar informações de proveniência.

Este capítulo aborda o estado da arte sobre Proveniência, incluindo definições, arquiteturas e a descrição de uma taxonomia de proveniência. As tecnologias envolvidas, inclusive a Web, são abordadas. Por fim, são descritos e comparados modelos de proveniência que utilizam ontologias e contribuem para interoperabilidade.

4.2 Definições

A etimologia da palavra Proveniência é o verbo francês *Provenir*, o qual significa Origem [194]. Na literatura, várias definições de proveniência têm sido propostas, porém, muitas são relacionadas a tecnologias específicas, não sendo diretamente aplicáveis à Web. O dicionário inglês Oxford [26] define *Provenance* com duas definições, conforme tradução a partir de [194]:

Definição 1. *O fato de vir de alguma fonte particular; origem, derivação.*

Definição 2. *A história ou pedigree de uma obra de arte, manuscrito, livro raro, etc.; concretamente, um registro da derivação e passagem de um item através de seus vários proprietários.*

A primeira definição relaciona-se com um conceito, denotando a fonte ou derivação de um objeto e a segunda se refere a um registro dessa derivação. Em sistemas computacionais, uma definição de proveniência é dada por [194]:

Definição 3. *Proveniência é definida como um processo, onde a proveniência de uma parte do dado é o processo que levou àquela parte do dado.*

Nesse sentido, atividades são realizadas pela execução de programas que apresentam entrada de dados, de estado e de configuração, produzindo saída de dados e de estado. Tais programas são compostos, de diversas formas, por programas mais simples, onde um processo é considerado como uma instância de uma execução, como uma computação.

A partir da perspectiva da Ciência da Computação, o objetivo é conceber uma representação de proveniência baseada em computador, permitindo análise e raciocínio lógico, necessitando capturar detalhes sobre os processos.

A proveniência definida como processo permite muitas informações serem capturadas, incluindo a derivação dos dados, bibliotecas utilizadas, hardware onde foi executada a computação e informações usadas em tempo de execução. Nesse caso, a proveniência necessita ser representada em um formato computacional para permitir raciocínio lógico. Outra definição envolve proveniência na Web, conforme a Definição 4.

Definição 4. *A proveniência de um recurso é um registro que descreve entidades e processos envolvidos em produzir e apresentar ou influenciar aquele recurso. Proveniência fornece um fundamento crítico para assegurar autenticidade, habilitar confiança e permitir reprodutibilidade. Asserções de proveniência são uma forma de metadados contextuais, podendo se tornar registros importantes, com sua própria proveniência.*

A quarta definição é apresentada pelo Grupo Incubador de Proveniência do W3C [50], onde proveniência deveria incluir informações sobre a criação e publicação de recursos da Web, assim como sobre acessos aos recursos e atividades para sua discussão, *linking* e reuso.

Pesquisas sobre proveniência envolvem diferentes comunidades, onde muitas definições alternativas de proveniência têm emergido na literatura, as quais são descritas na sequência, conforme [194].

- *Grafo Acíclico Dirigido (GAD)*: dessa forma é possível demonstrar graficamente como um produto de dado ou evento foi produzido em uma execução [195].
- *Why*: esse termo identifica tuplas existentes justificando um resultado de consulta [87].
- *Where*: ajuda a identificar de onde a informação vem, por exemplo, de um banco de dados [87].
- *How*: a definição *Why* não explica como tuplas estão envolvidas na criação de um resultado. A definição *How* exemplifica como uma tupla resultante é derivada, definida no contexto de álgebra relacional e *datalog* recursivo [100].

  Essas três definições de proveniência *Why*, *Where* e *How* surgiram inicialmente no contexto de banco de dados [87], as quais podem ser aplicadas a contextos mais amplos, como em *mashups*, onde a *Where*, por exemplo, pode ser *em cache* [194].
- *What*: caracteriza um evento, uma mudança de estado que acontece nos dados durante seu ciclo de vida [215].
- *When*: indica tempo ou, mais precisamente, a duração de um evento [215].
- *Who*: representa agentes, incluindo pessoas ou organizações envolvidas em um evento [215].
- *Which*: indica instrumentos ou programas usados em um evento [215].

  Estendendo as três definições de banco de dados, *Why*, *Where* e *How*, Ram e Liu [215] apresentam um modelo ontológico para proveniência de dados, denominado Modelo W7, o qual tem aplicabilidade na Wikipedia, adicionando à estas definições quatro elementos de proveniência: *What*, *When*, *Who* e *Which*, formando sete elementos interconectados.
- *Proveniência de Anotação*: nesse contexto, Dublin Core [9] fornece estrutura e semântica para metadados de recursos, tais como Autor, Data de Criação e Versão. Essa definição de proveniência é vista como uma especialização da Definição 3 em [194], relacionada a propriedades específicas de processos passados.
- *Proveniência de Serviço*: definição proposta por Mychlmayr et al [187]. Segundo Moreau [194], esta pode ser enquadrada na Definição 3 por considerar a proveniência do estado de um serviço, considerando interações anteriores e configuração inicial.

- *Workflow*: trilhar a proveniência de *workflows* tem sido uma preocupação crescente. Callahan et al [92] introduzem essa definição, onde *workflow* é um tipo específico de dado para o qual o processo de derivação necessita ser trilhado. Um exemplo é o ambiente VisTrails [92] que captura modificações feitas no *workflow* através de um ambiente de desenvolvimento integrado.

- *Proveniência Aproximada*: definição utilizada em bancos de dados probabilísticos [217], sendo uma alternativa para completar a proveniência, desconsiderando trilhar todas as derivações de proveniência, onde uma aproximação é considerada mais valiosa do que a proveniência precisa e completa retornada após longo tempo.

- *Proveniência de Interoperabilidade*: Moreau [194] apresenta essa definição como a habilidade de componentes acoplados expressarem e trocarem informações de proveniência pertencentes aos dados produzidos ou trocados, assim como permitem que sejam realizadas consultas sobre a proveniência.

- *Traço de Proveniência*: essa definição de proveniência é proposta por Cheney et al [99], onde um traço de proveniência tem propriedades associadas, tais como consistência e fidelidade.

- *Insight de Proveniência*: Gotz e Zhou [137] referem a um registro histórico dos processos pelo qual um *insight* é derivado durante uma tarefa visual analítica. Esse *insight* de proveniência é inferido a partir de interações de baixo nível do usuário em um *workflow*, onde é capturado automaticamente um registro semântico da atividade do usuário.

- *Proveniência Aumentada*: Chen et al [98] definem esse termo em um contexto de *workflow* para denotar a proveniência de um pedaço de dado, relacionando metadados semânticos sobre o processo que o originou.

- *Proveniência de Metadados*: estes em si são metadados, ou seja, dados sobre dados e deve ser possível republicá-los exatamente da mesma maneira e em conjunção com outras declarações de metadados descritos. O *Metadata Provenance Task Group* [9] objetiva definir um *profile* de aplicação permitindo fazer asserções sobre declarações de descrição. Neste grupo são criadas diretrizes de uso quanto a maneiras para conectar agregações de metadado - *Content* - com metadados de proveniência permitindo múltiplos processos de republicação.

- *Proveniência Semântica*: Sahoo et al [220] denotam proveniência na qual o conhecimento do domínio e descrições ontológicas estão relacionados. Segundo Moreau [194], essa definição é adotada por muitos sistemas e depende de especializar dependências de dados para domínios de aplicação específica.

- *Vocabulário de Proveniência*: Hartig [148] especializa o *Open Provenance Model* como um vocabulário de proveniência para descrever a proveniência de *Linked Data* sobre a Web.

Conforme Moreau [194], essas sete últimas definições de proveniência têm destaque no contexto da Web Semântica [73].

4.3 Arquiteturas

As principais arquiteturas de processamento de dados que envolvem proveniência são orientadas a dados e serviços [194]. Também existe a arquitetura orientada a *scripts* [225].

Moreau [194] e Simmhan [225] relatam que não há uma clareza quanto a essas divisões arquiteturais relacionadas a granularidade, pois mais ou menos detalhes podem ser fornecidos. Também, não existem duas soluções quanto a granularidade de proveniência, *coarse* x *fine*, e sim existem várias soluções, nas quais o fluxo de informações pode ser apresentado de modo mais detalhado ou de mais alto nível.

Da mesma forma, bancos de dados modernos permitem que serviços Web sejam invocados como parte de uma execução de consulta, assim como existem padrões para expor capacidades de bancos de dados através de interfaces de serviços. Ou seja, a distinção entre as arquiteturas de processamento de dados não é absoluta [225].

4.3.1 Arquitetura Orientada a Dados

Informações descrevendo como dados são transformados em banco de dados é referida como *fine-grain* ou proveniência *dataflow* [157].

Uma distinção na proveniência *fine-grain* é feita por Buneman et al [87], destacando os elementos de proveniência *Where* e *Why*. *Where* é a identificação dos elementos fonte de onde os dados do destino são copiados. Com *Why*, é feita a justificativa para o elemento aparecer no resultado.

Bancos de dados têm sido tradicionalmente utilizados para armazenar dados segundo algum critério lógico. A proveniência nesse contexto tem ampla variedade de aplicações.

Na década de 90, Wang e Madnick [242] desenvolveram o Modelo e a Álgebra denominada *Polygen (poly - múltiplas, gen - fontes*), para estudar sistemas de bancos de dados heterogêneos, objetivando endereçar questões tais como *De onde os dados vêm?* e *Quais fontes de dados intermediárias foram usadas para alcançar aquele dado?* Esse modelo utiliza proveniência como anotações, em cada coluna, de cada tupla. Os autores explicam que apesar da Álgebra Polygen não ter sido formalmente estudada, motivou trabalhos tais como [74].

Woodruff e Stonebraker [246] foram idealizadores da proveniência *fine-grain* quando resultados de consultas não são anotadas com proveniência. Esse trabalho permite ao programador definir inversões fracas para funções definidas em seu código. Uma inversão fraca, quando aplicada a algum elemento de dado, no resultado de uma função, retorna alguma aproximação para a proveniência que é associada com a função.

Cui e Widom [109] não utilizam a técnica de inversões fracas como em [246], mas analisam a estrutura algébrica de consultas relacionais, onde a proveniência é computada e agregada, conforme operadores algébricos usados na consulta.

Buneman et al [87] utilizam um modelo de dados em árvore, baseado em [109]. É apresentado um *framework* para consultas de proveniência de dados no contexto de Seleção, Projeção, Junção, União e Visões, usando as perspectivas *Why*, referindo-se aos dados fonte que tem alguma influência na existência dos dados e *Where*, referindo-se a localização de bancos de dados originais, a partir dos quais o dado foi extraído.

Cheney et al [100] expandem as formas *Why* e *Where* para gerar proveniência incluindo a forma *How*, objetivando explicar como um tupla resultante foi derivada de acordo com as consultas. São descritos os relacionamentos entre esses elementos de proveniência.

4.3.1.1 Coleções de Dados e *Streams*

Dados de proveniência podem estar representados como coleções de dados e *streams*, conforme apresentado em [194]. Uma coleção de dados é um grupo de itens de dados, geralmente homogêneos, podendo estar ordenados. Uma coleção de dados reflete propriedades de seus elementos, tais como todos os resultados produzidos por um experimento ou consulta, todas as simulações considerando um projeto, um conjunto de *Uniform Resource Identifiers* - URIs retornados por uma busca, entre outros. Pode variar com o tempo, ser persistida e recuperada.

Como exemplos de coleções de dados se destacam [194]: *sets* (diretórios de arquivos); relações (tabelas relacionais SQL); hierarquias (documentos XML) ou, *arrays* (matrizes dimensionais), entre outros. São entidades que podem ser anotadas, manipuladas, transformadas ou arquivadas. Moreau [194] explica que é importante distinguir a proveniência de uma coleção de dados de proveniência de seus membros individuais.

A proveniência de uma coleção abstrai os detalhes da proveniência de seus membros. Representar a proveniência de uma coleção e seus membros, segundo [194], é uma tarefa desafiadora, pois envolve alterações, granularidade e representação dos dados. Sistemas de *workflows* criam dados sem atualizar dados existentes, incluindo coleções, onde coleções de entrada dos processos mapeiam operações em elementos, produzindo novas coleções [194]. O sistema Kepler [63, 61] (Seção 4.3.2) permite que coleções sejam manipuladas, fornecendo operações para manipular os membros de uma coleção.

A arquitetura de banco de dados trata coleções, tais como: tabelas relacionais, linhas e células constituintes do modelo relacional; hierarquias em bancos de dados XML. A proveniência em banco de dados trata coleções de dados em vários níveis de granularidade, conforme [194]. Os elementos de proveniência *Why* e *Where* [87] fazem uso de dados semi-estruturados XML. Buneman et al [86] usam proveniência de conteúdos de células em atualizações, onde anotações para tabelas, linhas e células são propagadas. Cui e Widom [109] e Glavic [134] trabalham com tuplas na Linguagem SQL, em ambiente de *data warehouse* e no sistema PERM, respectivamente. O *Open Provenance Model* - OPM apresenta *collections profile* [196], permitindo a proveniência de membros ser derivada a partir da proveniência de coleções, aplicando regras de inferência específicas da operação feita na coleção.

Streams são coleções temporais que nem sempre são persistidas, sendo de interesse tanto da comunidade de bancos de dados quanto de *workflows*. *Streams* apresentam problemas específicos quanto a proveniência. Exemplos de consultas envolvendo *streams* são: *De qual sensor essa parte do dado foi construído?* e *Quais transformações foram envolvidas na transformação do stream?*, entre outras, conforme citado em [194].

Várias técnicas têm sido desenvolvidas para proveniência em *streams*, onde tais dados são usados em ferramentas de simulações, modelagem e análises. Moreau [194] relata alguns exemplos, como segue. O sistema de *workflow* Kepler é relacionado a aplicações *streaming*, onde Anand et al [63] propõem um modelo de proveniência considerado em si um *stream* inserido em aplicações desse tipo. Misra et al [191] descrevem uma abordagem onde o sistema recria o grafo de processamento que gerou a saída, fornecendo elementos de *streams* de dados intermediários que o geraram.

4.3.1.2 Uso de Semântica em Bancos de Dados

Muitos trabalhos desenvolvidos envolvendo proveniência em bancos de dados incluem semântica, como as abordagens de proveniência implícita, formalismos, traços de proveniência, expressividade em modelos de dados de proveniência, propagação de informações em consultas e subconsultas, entre outros. Moreau [194] descreve alguns exemplos como segue.

A proveniência implícita considera a semântica da linguagem de consulta, sendo proposta por Buneman et al [86], onde um valor é descrito por anotações, referido como um esquema de cores, denotando a origem do mesmo. Para qualquer resultado produzido por um programa, anotações associadas indicam de onde o valor foi derivado.

Souilah et al [229] formalizaram a proveniência em sistemas distribuídos baseados em *Pi-calculus*. Todos os produtos de dados são anotados com metadados, representando sua proveniência. As anotações são sequências de eventos enviados e recebidos, estendidas sempre que valores são comunicados pela aplicação. Nesse caso é verificado se o que é dito sobre o passado de um valor corresponde ao seu atual valor.

Um traço de proveniência, segundo Cheney et al [99] apresenta propriedades, tais como consistência, descrevendo o que aconteceu durante sua execução e fidedignidade, para afirmar se o traço de proveniência contém informação suficiente para descrever como o programa deveria ter executado com diferentes entradas.

Moreau [194] formalizou o *Open Provenance Model*, citado como uma língua franca para interoperabilidade. Outros modelos de proveniência são abordados na Seção 4.6.

Quanto a propagação de informações em consultas e subconsultas, destacam-se os trabalhos que seguem. Tan [234] classifica sistemas que trabalham com anotações como atualização de proveniência *eagerly*, onde a informação é propagada em tempo de execução, e os que trabalham com funções e/ou transformações como *lazy*, ou seja, atualização é gerada sob demanda, por meio de uma consulta.

Bhagwat et al [74] propõem um sistema de gerenciamento de anotações para bancos relacionais, com anotações do tipo *Where* sendo propagadas. DBNotes se destaca nesse contexto [101], fazendo uso de anotações em bancos de dados relacionais. Quando uma consulta é executada, notas de valores de atributos relevantes no banco de dados são automaticamente propagadas para valores de atributos no resultado da execução da consulta, caracterizando uma abordagem *eagerly* de geração de proveniência.

Cui e Widom [109] definem a proveniência de tupla relacional em uma visão, sendo gerada uma consulta inversa para recuperar todas as combinações de tuplas-base que satisfazem essa definição. Caracterizando uma abordagem *lazy* de geração de proveniência com aplicabilidade em ambiente de *data warehouse*, evoluindo para a proveniência *Why*.

Glavic e Alonso [134] demonstram que usando subconsultas, como em [109], a proveniência pode incluir tuplas que não contribuem nos resultados [134]. Esses autores propõem o sistema *Provenance Extension of the Relational Model* - PERM, estendendo o PostgreSQL, com a reescrita de consultas SQL para propagar proveniência junto com os resultados de consulta. Essa reescrita de consultas e seus resultados, usando o mesmo modelo de dados, pode ser acessada, armazenada e otimizada com técnicas tradicionais de bancos de dados relacionais.

Por outro lado, SPIDER [60] é uma abordagem que não usa anotações, fazendo uso de mapeamentos de esquemas, asserções lógicas de relacionamentos entre uma instância de um esquema fonte e uma instância de um esquema destino. A semântica de mapeamentos é feita através de descrições de informações de proveniência.

4.3.2 Arquitetura Orientada a Serviços

Informações descrevendo como dados são calculados a partir de um fluxo de observações, são referidas como *coarse-grain* ou proveniência *workflow* [157]. Amplo uso de ferramentas de *workflow* para processamento de dados científicos facilita a captura de proveniência. O processo de *workflow* descreve todos os passos envolvidos na produção de um conjunto de dados, capturando informações de proveniência.

Tan [234] relata que a proveniência em um *workflow* refere-se ao registro do histórico de derivação da saída do *workflow*, onde a quantidade de informações registradas para proveniência pode variar. Pode incluir um registro completo das sequências de passos realizadas em um *workflow* em um conjunto de dados, incluindo um registro detalhado das versões de software, assim como modelos e equipamentos de hardware utilizados, permitindo identificação de passos que não necessitem ser repetidos, acerca de diferentes execuções.

O *workflow* captura o fluxo de dados e de controle lógico, entre diferentes serviços, podendo ser representado como um grafo direto, com serviços formando nodos e parâmetros de configuração de produtos de dados e serviços, as arestas. *Workflows* são geralmente usados para composição de serviços, numa arquitetura orientada a serviços, comumente implementadas como serviços Web ou de grade, ou ainda como *scripts*. Como exemplos citam-se Taverna [254], Pegasus [188], VisTrails [92] e Kepler [63]. Tan [234] categoriza nesse caso a proveniência como *coarse-grain*. Em contraste, proveniência *fine-grain* explica como os dados foram derivados.

Os *workflows* podem ser escritos em linguagens, tais como *Web Services Flow Language* - WSFL [171] ou *Business Process Execution Language* - BPEL [64], as quais são executadas e gerenciadas por um motor de *workflow* - *workflow engine* que controla invocações [225]. Algumas aplicações de *workflows* são descritas na sequência.

- Wings/Pegasus

  Kim et al [163] propõem o sistema Wings/Pegasus, o qual faz uma separação entre desenvolvimento da aplicação e execução, afim de gerenciar os processos. *Workflows* que capturam o comportamento da aplicação são abstratos, em nível da aplicação, descrevendo os componentes da aplicação e suas dependências. Pegasus mapeia esses *workflows* abstratos, descritos de forma independente de recursos, para recursos apropriados, heterogêneos e distribuídos, dependendo da disponibilidade e características dos recursos da rede [157].

  Pegasus faz uso de um *template* reusável, instanciado para conter detalhes de execução [194]. O compilador tem como entrada um *workflow* abstrato, especificado como um grafo acíclico dirigido, compilado como um *workflow* instanciado, executável por um *workflow engine*, onde a localização da computação, transferências de dados e bibliotecas invocadas tornam-se explícitas [188].

  A proveniência é estruturada de modo similar em [163], registrando ações para cada refinamento e passo de execução. Refinamento do *workflow* e documento de execução habilitam a proveniência detalhada de um item de dado ser encontrada, permitindo aos usuários encontrarem passos executáveis do *workflow* para cada item de dado e, também, conexão entre *workflow* abstrato e executável. Porém, Islam [157] comenta que registrar cópias de todos os dados passando pelo sistema é impraticável, onde existe uma necessidade de técnicas adicionais para cópias com

conjuntos de dados muito grandes, assim como para consultas de proveniência, permitindo que usuários encontrem informações de interesse.

Esse sistema é um exemplo de proveniência, podendo ser usada para entendimento, em tempo de execução do sistema de *workflow*, conectando informações em tempo de execução para a especificação abstrata original, conforme projetado pelo usuário.

- VisTrails

  Nem sempre um *workflow* trabalha de modo automatizado, pois há necessidade de interação humana, onde a proveniência é importante. Groth el al (2006) [140] reconhecem interações do usuário e anotações, as quais podem ser exploradas por outros usuários para descobrir recursos.

  VisTrails [92] é um *workflow* científico desenvolvido em Python, multiplataforma, *open-source* e um sistema de gerenciamento de proveniência que fornece suporte para simulações, exploração de dados e visualização. Utiliza um modelo de proveniência baseado em ações, capturando mudanças em parâmetros e assegurando que usuários são capazes de reproduzir visualizações. Registra a história de explorações do usuário, em ambientes de visualização, aumentado com a capacidade para usuários anotarem suas explorações. São geradas visualizações interativas de proveniência de dados usando uma técnica espaço-temporal.

  O que difere VisTrails é uma infra-estrutura de proveniência abrangente, a qual mantém informações detalhadas sobre a história dos passos seguidos e os dados obtidos no decorrer de uma tarefa exploratória. Esse sistema mantém proveniência de produtos de dados, dos *workflows* que derivaram tais produtos e suas execuções. Essas informações são mantidas em arquivos XML ou em bancos de dados relacionais, permitindo aos usuários navegar em versões de *workflows* de forma intuitiva e desfazer alterações sem perder resultados. Permite a criação e execução de *workflows*, combinando recursos de baixo acoplamento, bibliotecas especializadas, grades e serviços Web.

- Taverna/myGrid

  Taverna [254] é um sistema de *workflow* de dados e serviços em bioinformática, desenvolvido pelo Projeto myGrid. Seu modelo de proveniência captura tanto a proveniência interna, gerada localmente, quanto externa, obtida a partir de provedores de dados. A Web Semântica de proveniência Ouzo combina esses diferentes tipos de proveniência, fazendo uso de anotações semânticas. Ouzo é representado em RDF/RDFS. Possui um componente de consulta de proveniência denominado *Provenance Query and Answer* - ProQA, o qual dá suporte à recuperação de proveniência como abstração de proveniência, agregação e raciocínio semântico. ProQA é implementado como um conjunto de APIs, podendo ser utilizadas como serviços de proveniência, para compor *workflows* de proveniência do sistema, as quais analisam resultados de experimentos usando registros de proveniência.

- Kepler

  O sistema de *workflow* Kepler [63] é construído com base em uma abordagem orientada a ator, baseado na Linguagem Java. Auxilia na criação de *workflows* e execução de processos, onde cientistas podem projetar, executar, monitorar, re-executar e comunicar passos analíticos de forma fácil. Mantém a trilha de todos os tipos

de proveniência, sendo na execução do *workflow*, proveniência de dados e processos e armazenamento eficiente e uso de dados [157].

Kepler facilita a modelagem e projeto de sistemas complexos, contribuindo na composição de *Web services*. Dá suporte a diferentes tipos de dados e computa aplicações intensivas, variando a partir de aplicações analíticas locais, a *pipelines* distribuídos de alto desempenho e processamento. Uma desvantagem desse sistema é que usuários não podem desenvolver *workflows* de forma independente de recursos, como em Pegasus, pois não fornece benefícios de portabilidade do *workflow*, otimização e facilidade de projeto.

4.3.3 Arquitetura Orientada a *Scripts*

Outra arquitetura de processamento de dados, citada por Simmhan [225], é a arquitetura de *scripts*. Usualmente, cientistas escrevem *scripts* para uso com aplicações científicas, as quais podem ser escritas em várias linguagens de programação, tais como FORTRAN, C, C++, and Java. *Scripts* se caracterizam como um modelo de programação modular, fornecendo interfaces padrão para desenvolvimento e reuso de tarefas científicas.

Entradas e saídas de *scripts* são feitas através de parâmetros, podendo ser em linha de comando ou carregados a partir de uma lista de arquivos contendo argumentos. Ambientes de desenvolvimento de *scripts* incluem ferramentas, tais como Matlab, J/Python, Tcl, e Perl, as quais fornecem poderosas bibliotecas que habilitam tarefas avançadas como interação com banco de dados, movimentação de arquivos e interfaces com recursos de rede.

Com *scripts* é possível agrupar tarefas para gerenciar aplicações complexas, como um sistema de *workflow*. Um *script workflow*, ou até mesmo uma aplicação separada de *workflow*, pode invocar vários *scripts* em sucessão, para executar experimentos científicos. O fluxo de dados entre *scripts* é acompanhado por passar referências para arquivos de conjuntos de dados como parte de parâmetros do *script* ou, inserir a localização do conjunto de dados dentro do código [225].

4.4 Taxonomia

Simmhan et al [226] apresentam uma Taxonomia de Proveniência baseada no registro de proveniência, seu conteúdo, representação e armazenamento, assim como formas de recuperar essas informações.

A Taxonomia de Proveniência é dividida em cinco principais categorias (Figura 4.2): Aplicações, Orientação e Granularidade, Representação, Armazenamento e Disseminação, as quais são discutidas na sequência.

4.4.1 Aplicações

Conforme a Taxonomia de Proveniência (Figura 4.2), o uso de proveniência está relacionado às seguintes aplicações: qualidade dos dados, trilha de auditoria, replicação, atribuição e informacional.

- *Qualidade dos dados:* informações de proveniência podem ser usadas para estimar a qualidade e confiabilidade dos dados, baseadas na origem e transformação dos dados [160]. Seu uso, juntamente com metadados, estima a qualidade dos dados para os

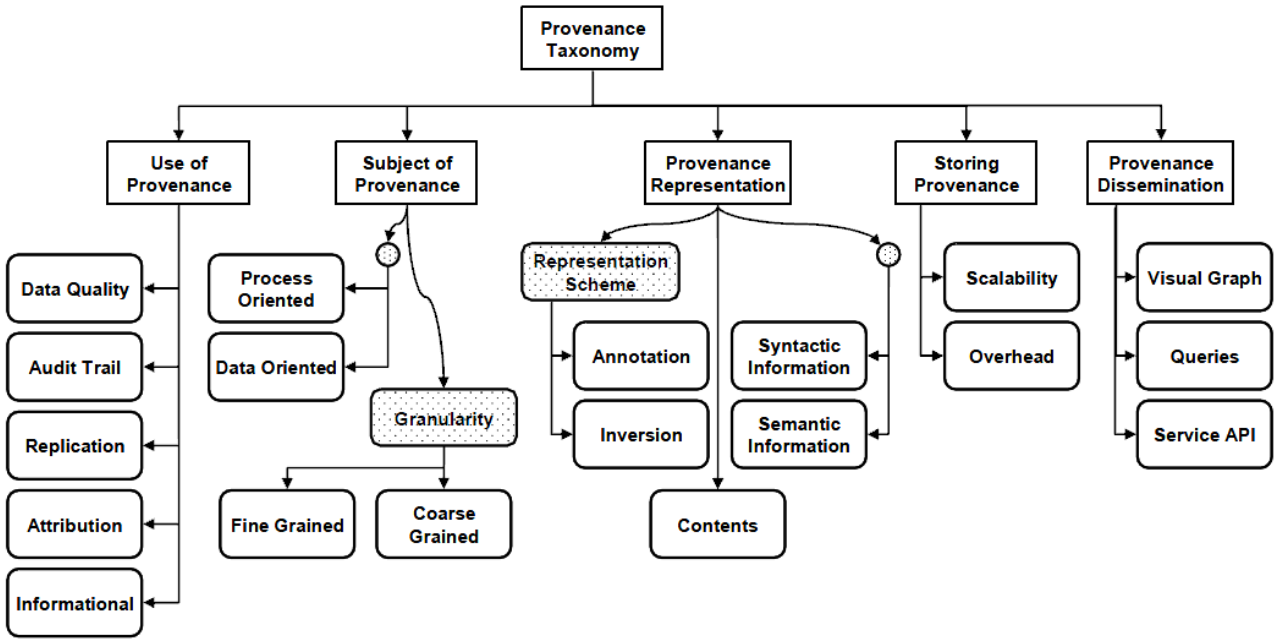


Figura 4.2: Taxonomia de Proveniência [226].

usuários, como forma de garantia de sua proveniência [176]. Pode também fornecer declarações de provas na derivação dos dados [110].

Simmhan et al [226] afirmam que a proveniência sobre um conjunto de dados permite avaliar a qualidade de suas aplicações. A qualidade dos dados de origem é importante, uma vez que erros gerados por falhas nos dados tendem a expandirem para dados derivados a partir deles. O nível de detalhe, incluído na proveniência, determina a extensão para a qual a qualidade dos dados pode ser estimada.

- *Trilha de auditoria:* proveniência pode ser usada para informar a trilha de auditoria dos dados, determinando o uso de recursos e detecção de erros na geração dos dados, onde são utilizados *logs*.

  Além de auditoria nos dados e processos pelos quais foi produzida, a proveniência pode otimizar o processo de derivação e coleta de estatísticas para descrever usos de recursos [139]. A proveniência na forma de traços em tempo de execução pode ajudar na verificação da ocorrência de exceções na criação dos dados.

- *Replicação:* esse caso envolve a proveniência detalhada, incluindo passos usados para derivar um conjunto particular de dados, repetição de experimentos, comparações, recriações, replicações ou atualizações parciais, entre outras. Se a proveniência contém detalhes suficientes das operações, fontes de dados e parâmetros, pode ser possível repetir a derivação, como em [129].

- *Atribuição:* envolve questões de direitos autorais e propriedade dos dados, citações, verificações de usuários dos dados, entre outros. Citações [160] são uma parte importante da publicação, principalmente em *eScience*, onde a proveniência age como uma forma de citação na publicação de conjuntos de dados científicos.

- *Informacional:* um uso genérico de proveniência é em consultas baseadas em metadados para descoberta de dados. Essa categoria pode ser navegável como uma árvore

de derivação ou outras formas gráficas para interpretação dos dados, explorando também outros metadados sobre os dados e processos.

Anotações utilizadas juntamente com proveniência podem ajudar a interpretar dados, especialmente para dados arquivados que são utilizados, posteriormente à sua geração, contribuindo para assimilar dados, de forma não ambígua, no domínio da aplicação do usuário [225].

Além destas, Buneman e Tan [89] identificam outras aplicações de proveniência:

- *Bancos de dados curated*: são bancos de dados científicos, manualmente editados, gerados por cópias, correções, anotações e integração de dados, a partir de outras fontes. Devem apresentar alta qualidade, pois estão se tornando uma forma de publicação científica reconhecida. Nesse caso, a proveniência é um componente essencial para assegurar a qualidade dos dados.

  Esses bancos não são, simplesmente computados por consultas, possuindo também valor adicionado na forma de correções e anotações por especialistas. O trabalho de Buneman et al [85] se destaca nesse contexto, onde cientistas, manualmente, assimilam conteúdo sobre um tópico especializado, a partir de várias fontes, sendo resultado de um grande tratamento de anotações manuais, correções e transferência de dados. Muitos desses bancos agem como publicações e estão atualmente substituindo trabalhos de referência tradicionais, tais como dicionários, enciclopédias, entre outros.

  Em bancos *curated*, elementos de dados são copiados, com frequência, a partir de um banco para outro. Um conhecimento de onde - *where* um elemento de dado vem é essencial na avaliação da qualidade do banco de dados [89]. Nesse caso, a proveniência sobre a criação, atribuição ou versão dos dados, permite assegurar a integridade e o valor científico dos mesmos.

- *Anotações:* caracteriza o processo de adicionar ou marcar dados existentes. A propagação de anotações é baseada em proveniência, podendo ser propagada a partir da origem para a fonte [74], assim como vice-versa.

- *Bancos de dados probabilísticos:* conhecidos como bancos de incertezas. A proveniência é usada para determinar se as tuplas nos resultados de uma consulta são independentes. Possui aplicabilidade no Projeto Trio [245]. A proveniência aproximada para bancos probabilísticos também é tratada em Ré et al [217], destacando que é frequentemente desnecessário trilhar todas as derivações, pois uma aproximação, retornada mais rapidamente, é mais valiosa do que a proveniência precisa e completa retornada após longo período de tempo. A proveniência aproximada é proposta como uma alternativa para que a mesma seja completada.

- *Compartilhamento e Integração de Dados:* Orchestra [251] é um sistema de compartilhamento de dados colaborativo que descreve política de confiança. Proveniência é utilizada para descrever relacionamentos entre a origem e o destino em um cenário de integração, objetivando o entendimento e *debugging* da especificação do sistema de integração.

- *Confiança:* em bancos de dados científicos, proveniência é importante para assegurar confiança nos dados. Green et al [138] propõem uma abordagem *semirings*, a qual

modela proveniência como *semirings* de polinômios para descrever a confiança nos dados, sendo útil em *reasoning* sobre manutenções de visões recursivas.

4.4.2 Orientação do Modelo e Granularidade

A orientação do modelo pode ser aplicada aos dados ou processos [255], conforme a Seção 4.3. A granularidade diz respeito às formas *fine-grained* e *coarse-grained*. A orientação e a granularidade de proveniência dizem respeito à questão: *Qual é a proveniência sobre isso?*

Os múltiplos níveis de detalhes para obtenção da proveniência em relação à granularidade relacionam-se com sua utilidade conforme o domínio. A granularidade varia a partir de atributos e tuplas em bancos de dados que representam *pixels* individuais [246], ou elementos de *arrays* [178] para coleções de arquivos [209], entre outros. O uso crescente de conjuntos de dados abstratos [129, 255] que se referem a dados em qualquer granularidade ou formato, permitem uma abordagem flexível, tornando a coleta de proveniência independente da granularidade ou representação do conjunto de dados. O custo de coletar e armazenar proveniência pode ser inversamente proporcional a sua granularidade [226].

4.4.3 Representação

A representação de proveniência envolve o esquema, o formato e o conteúdo de representação da proveniência.

- *Esquema de Representação:* diferentes técnicas podem ser usadas para representar a proveniência. A maneira na qual a proveniência é representada tem implicações no custo de registro e na riqueza de seu uso. As duas maiores abordagens de esquemas de representação de informações de proveniência usam anotações ou inversões de consultas [226, 225].

  - *Anotações:* neste caso, metadados compreendendo a história de derivação de um produto de dados é coletado como anotações e descrições sobre a origem dos dados e processos. A proveniência é pré-computada e prontamente usável como metadados [74]. Com anotações, dados antecedentes e os passos usados para sua derivação podem ser representados como um grafo acíclico dirigido. Esta forma de geração de proveniência requer mais espaço de armazenamento, porém são mais ricas, segundo os autores, em relação à história de derivação, do que a inversão de consultas.
  - *Inversão de Consultas:* no método de inversão de consultas, algumas derivações podem ser invertidas para encontrar a entrada de dados fornecida para sua derivação, sendo utilizadas consultas e funções definidas pelos usuários em bancos de dados, as quais podem ser invertidas, automaticamente, ou por funções explícitas [246, 109, 245]. Informações de proveniência sobre as consultas e os dados resultantes podem ser suficientes para identificar a fonte dos dados. Essa é uma abordagem mais compacta do que usar anotações, entretanto, nem todas as funções são inversíveis, onde dados auxiliares podem ser necessários, assim como são restritas a determinadas classes de consultas relacionais, não sendo universalmente aplicáveis. Também, nem todas as funções definidas por usuários têm funções inversas. Nesse caso, a proveniência é tratada *just-in-time* [226].

- *Formato:* o formato de proveniência pode ser descrito sintática ou semanticamente:
  - *Informação Sintática:* nesse caso geralmente a linguagem XML é adotada para representar proveniência [188, 255, 199, 77, 195]. Também muitos sistemas usam a arquitetura orientada a serviços, onde XML é o formato principal para troca de mensagens.

    Implementações específicas para inversões de consultas dependem do formato da consulta, por exemplo, SQL em bancos de dados relacionais ou funções definidas por usuários para processar os dados.
  - *Informação Semântica:* a estrutura semântica relaciona-se com o conhecimento semântico sobre a linguagem semântica usada para definir metadados de proveniência. Nesse caso, ontologias modelam conceitos e relacionamentos usados na geração da proveniência, favorecendo inferências para descoberta de conhecimento implícito, por meio de linguagens como RDF e OWL [255, 199].

    A semântica proporciona como vantagens a descrição do contexto, melhoras em resultados de pesquisas e provas de origem. Faz uso de ontologias para descrição do conhecimento semântico, onde existem esforços para promover interoperabilidade dos dados gerados [195, 182, 220].

- *Conteúdo:* Simmhan et al [226] afirmam que informações de proveniência relacionadas ao método de inversão de consultas são esparsas e limitadas, armazenando a consulta e/ou processos que criaram os dados derivados, tais como *Por quê* - (*Why*) os dados foram criados, servindo para identificar a origem dos dados que criaram os dados derivados, assim como *Onde* - (*Where*) eles se originaram. Anotações, por outro lado, podem ser mais ricas e, em adição ao histórico de derivação, pois podem incluir parâmetros passados para os processos de derivação, as versões de *workflows* que permitirão a reprodução dos dados, ou, até mesmo, podem estar relacionados às referências de publicação. Anotações podem ser suficientes para repetir o processo de derivação ou reproduzir os dados derivados. Segundo os autores, trata-se de um ponto discutível, onde o limite entre informações de proveniência e o uso de metadados genéricos estão situados. Em alguns casos, há pouca distinção entre as duas formas, onde a proveniência é, em geral, gerenciada por meio de metadados.

  Atualmente, não há um padrão de metadados para representação de proveniência acerca de muitas áreas, devido às suas necessidades diversas. Todavia, Simmhan [225] afirma que existem esforços para desenvolver tais padrões, como [195, 199].

4.4.4 Armazenamento

O armazenamento de proveniência diz respeito se os dados de proveniência serão armazenados de forma conjunta ou separada dos dados. É um fator crucial, visto que o armazenamento de proveniência pode vir a ser maior que os dados. Também, a forma como metadados de proveniência são armazenados, influencia a escalabilidade [68].

Anand et al [63] propõem uma representação compacta de proveniência, associada com regras de inferência, permitindo que todas as dependências sejam explicitamente derivadas. Esse sistema rastreia alterações feitas em estruturas de dados, assumindo que qualquer outro elemento permaneça idêntico.

O armazenamento de proveniência pode ser fortemente acoplado com os dados descritos e localizados no mesmo sistema de armazenamento ou, embutidas dentro dos arquivos de

dados. Para exemplificar este último caso, cita-se as imagens científicas do tipo *Flexible Image Transport System* - FITS. A Especificação FITS [146] estabelece regras relacionadas a esse tipo de imagem, a qual difere de formatos tradicionais de imagens tais como *Joint Pictures Expert Group* - JPEG, *Graphics Interchange Format* - GIF, *Portable Network Graphics* - PNG, entre outros, devido a sua estrutura básica que é formada por um cabeçalho - *header* contendo metadados, tais como SIMPLE, BITPIX, NAXIS, EXTEND, COMMENT, HISTORY, entre outros, e uma matriz usada para armazenar dados binários. A Figura 4.3 mostra um sumário deste *header*, obtido a partir da biblioteca Java fits.jar [15].

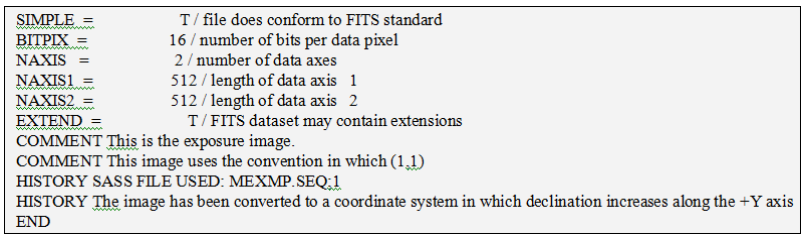


Figura 4.3: Sumário de um *header* FITS [146].

Entretanto, a especificação FITS não contempla a adição de metadados de proveniência, descrevendo o uso do metadado HISTORY para armazenar passos executados. Essa forma de geração de proveniência permite manter a integridade. Porém, é texto livre, não sendo legível por máquina, dificultando sua utilização por agentes computacionais. Também, são difíceis para publicar e pesquisar sobre a proveniência [225]. A escalabilidade e sobrecarga de proveniência são abordadas na sequência.

- *Escalabilidade:* relaciona-se com conjuntos de dados, nível de proveniência, granularidade, distribuição geográfica, usuários, esquema de representação e as formas de armazenamento distribuído ou centralizado.

  O método de inversão de consultas é mais escalável do que o método utilizando anotações. Entretanto, é possível reduzir a necessidade de armazenamento no método de anotações por registrar justamente o passo de transformação, imediatamente precedente que criou os dados e, de forma recursiva, inspecionar a proveniência de seus passos antecedentes para completar o histórico de derivação.

- *Sobrecarga:* a sobrecarga está relacionado com o armazenamento e automação dos dados. O gerenciamento de proveniência envolve custos para sua coleção e armazenamento. Barga e Digiampietri [68] e Scheidegger et al [221], propõem um modelo de proveniência em camadas para armazenar traços de proveniência em um gerenciador de armazenamento, resultando em uma representação que evita redundância dos dados.

  Informações de proveniência menos utilizadas podem ser arquivadas para reduzir a sobrecarga de armazenamento, assim como pode-se manter a proveniência para o que for mais utilizado. Quando a coleta de proveniência é feita manualmente, isso pode impedir que a proveniência seja completamente gerada e disponível para legibilidade por máquina [255].

  A sobrecarga de manter múltiplas versões de um registro de dados, em bases de dados científicas, na presença ou não de compressão, é discutida por Buneman e

Tan [88]. Em bancos de dados *curated*, Buneman et al [85] investigam os requisitos de armazenamento, associando a sobrecarga para seu modelo de proveniência *copy-and-paste*.

Chapman et al [96] também investigam o problema de aumento de requisitos de armazenamento para proveniência em *workflows*, por meio das técnicas de processos de fatorização e baseadas em herança. No primeiro caso, utiliza sub-expressões na proveniência de diferentes itens, permitindo-lhes serem armazenadas uma vez em cada item. No segundo caso, quando a proveniência pode ser herdada por um item, o mecanismo de herança pode corretamente instanciar o que é requerido, reduzindo requisitos de armazenamento, permitindo que a proveniência seja consultável.

A manutenção diz respeito à mutabilidade e versionamento dos dados, assim como o responsável por sua geração. A mutabilidade diz respeito à atualização ou se versões serão criadas. O mecanismo de coleção de proveniência e seu repositório de armazenamento determinam a confiança na origem dos dados, sendo necessários serviços de mediação [188].

4.4.5 Disseminação

Segundo a Taxonomia de Proveniência (Figura 4.2), a disseminação envolve visualizações gráficas, consultas e serviços. Uma forma comum de disseminar proveniência é por meio de um grafo de derivação, sendo possível navegar e inspecionar utilizando uma interface gráfica [255, 199, 77]. Também é possível pesquisar conjuntos de dados utilizando metadados de proveniência [220]. A disseminação de proveniência possibilita verificar como os dados foram criados, pela reconstituição ou declarações de provas, conforme [182].

Existem várias tecnologias para consultas de proveniência em [194], tais como SQL, XQuery, Xpath, SPARQL, entre outras. Porém, linguagens específicas a um domínio têm sido projetadas para recuperar proveniência, melhorando a expressividade por oferecer novos construtores e abstrações, facilitando a formulação de consultas complexas. Moreau [194] descreve algumas dessas tecnologias, conforme segue.

Um exemplo que se destaca é a interface de consulta para proveniência PASOA - *Provenance Aware Service Oriented Architecture* [140]. Essa interface faz uso de consultores para identificar itens de dados que se deseja saber a origem e de uma especificação de parte do processo para o qual tais consultores têm interesse em obter uma descrição.

PASOA [140] identifica objetos de forma intensional, considerando passos do *workflow*, por exemplo, o objeto contido em uma coleção, enviado como entrada de um passo do *workflow* e executado por um serviço específico. A interface de consulta PASOA permite que tais descrições sejam especificadas como uma expressão XPath sobre o conjunto de asserções de processos. Essa abordagem oferece várias formas de especificar o escopo do processo, podendo ser delimitada pela localização, por tipos de derivação ou pelo tipo de dados intermediários envolvidos na computação.

Determinados sistemas fazem uso de um identificador único para todos os resultados intermediários, utilizados para obter sua proveniência. Taverna/myGrid usa *Life Science IDentifiers* - LSIDs [254]. Swift usa *tags* URIs [104]. Essa abordagem é caracterizada como extensível, pois itens de dados são numeradas de forma explícita.

A Linguagem de Consulta para Proveniência - *Query Language for Provenance* - QLP [63] foi projetada para ser independente da representação física, incluindo construtores

específicos para consultas de proveniência em *workflows* científicos. Consiste dos componentes consultas *lineage*, objetivando percorrer o fechamento transitivo do grafo e, estrutural, objetivando percorrer coleções aninhadas, onde ambas podem ser combinadas em consultas híbridas.

A linguagem de consulta de proveniência de VisTrails - vtPQL faz uso da estrutura de proveniência em camadas deste *workflow*, onde cada nível da consulta é uma simples expressão SQL com funções adicionais, predicados e atributos [221] .

Na comunidade de banco de dados, são definidas consultas do tipo *Why Not*, sendo propostos dois algoritmos para consulta, percorrendo o grafo de proveniência em ambas direções [96]. Esse tipo de consulta produz uma série de declarações sobre as razões potenciais que os dados de interesse dos usuários estão faltando, a partir de um resultado gerado. A resposta é que modificações para um banco de dados existente deveria ser requerida para uma não-resposta, como uma tupla faltante, para um resultado de consulta tornar-se uma resposta.

4.5 Proveniência na Web e a Web semântica

Moreau [194] define fundamentos para proveniência na Web, a qual habilita reprodutibilidade de resultados científicos, necessária para trilhar informações em bancos de dados *curated* [85]. A proveniência é caracterizada pelo autor como essencial para que raciocinadores lógicos realizem julgamentos de confiança sobre informações usadas na Web Semântica [73], afirmando que *"A sociedade pode e deveria confiavelmente trilhar e explorar proveniência de informações na Web"*.

Múltiplas fontes de dados têm sido usadas para compilar a maior base de dados bibliográfica em proveniência, permitindo analisar tendências emergentes na comunidade científica [194].

Segundo Moreau [194], o banco de dados de bibliografia de proveniência contempla uma lista dos cinquenta *papers* mais citados dentro do banco de dados. A primeira publicação data de 1986 [69], descrevendo uma técnica de autoria para ajudar analistas a entenderem e validarem resultados de dados. O artigo *Where and Why Provenance* [87] ocupa o topo da lista, seguido por dois *surveys*, segundo [194] (Figura 4.4).

Rank	Citations	Paper (first author:venue)	Rank	Citations	Paper (first author:venue)
1	(116)	Buneman:ICDT01	26	(27)	Frew:SSDBM01
2	(97)	Simmhan:SIGMOD05 (*)	27	(27)	Bowers:IPAW06
3	(65)	Foster:SSDBM02	28	(24)	Simmhan:ICWS06
4	(62)	Cui:TODS00	29	(24)	Moreau:OPM1.00
5	(56)	Bose:ACMCS05 (*)	30	(24)	Benjelloun:VLDB06
6	(49)	Moreau:CCPE08	31	(24)	Buneman:FSTTCS2000
7	(47)	Woodruff:ICDE97	32	(23)	Barga:CCPE08
8	(47)	Miles:JOGC07	33	(23)	Greenwood:AHM03
9	(44)	Groth:D3.1.1	34	(23)	Zhao:SWT03
10	(40)	Bhagwat:VLDB04	35	(20)	Wang:VLDB90
11	(39)	Muniswamy-Reddy:USENIX06	36	(19)	Moreau:CACM08
12	(38)	Widom:CIDR05	37	(18)	Buneman:ICDT07
13	(38)	Buneman:SIGMOD06	38	(18)	Lanter:CGIS91
14	(37)	Freire:IPAW06	39	(18)	Bavoil:VC05
15	(36)	Groth:OPODIS04	40	(18)	Bowers:CCPE08
16	(36)	Cui:VLDB03	41	(18)	Simmhan:IPAW06
17	(33)	Fosterb:PROV02	42	(17)	Miles:CCPE08
18	(31)	Zhao:CCPE08	43	(17)	Zhao:IPAW06
19	(31)	Groth:HPDC05	44	(17)	Braun:IPAW06
20	(30)	Szomszor:ODBASE03	45	(17)	Myers:SWT03
21	(30)	Cui:ICDE00	46	(16)	Kim:CCPE08
22	(29)	Zhao:ISWC04	47	(16)	Zhao:ICSNW04
23	(29)	Altintas:IPAW06	48	(16)	Frew:CCPE08
24	(28)	Moreau:IPAW06 (*)	49	(16)	Tan:DBBUL07(*)
25	(27)	Green:PODS07	50	(16)	Groth:AHM05

Figura 4.4: Publicações mais citadas sobre proveniência [194].

A literatura sobre proveniência contempla muitos *surveys* nos últimos dez anos, conforme [194, 133]: uma revisão de proveniência em *e-Science* [226]; proveniência para processamento

científico [77]; proveniência em bases de dados [100]; classificação de abordagens [135]; fundamentos para proveniência na Web [194]; confiança em Ciência da Computação e a Web Semântica [65] e proveniência em tarefas computacionais [130].

Entretanto, a maioria dos trabalhos sobre proveniência envolve bancos de dados, *workflow* e pesquisas em *e-Science* [194]. Sistemas de bancos de dados e *workflows* são sistemas fechados, ou seja, os sistemas de gerenciamento têm controle completo nos dados que gerenciam, trilhando a proveniência somente dentro de seu próprio escopo. Na Web, uma abordagem mais ampla é requerida, onde a representação de proveniência precisa considerar múltiplos sistemas.

Aplicações tradicionais apresentavam características de ser monolíticas, executar dentro de uma simples máquina, requerendo mínima segurança, e não necessitando interoperar com outros sistemas. Atualmente, aplicações requerem necessidades diferentes, tais como: consistem de muitos componentes, envolvem várias tecnologias, estão em diferentes domínios de segurança, fazem uso de orientação a serviços, possuem fraco acoplamento e reuso, além de executarem na Web. Nesse contexto, Moreau [194] afirma que o desafio é trilhar proveniência de dados acerca de múltiplas tecnologias e domínio de segurança, envolvidos em sua derivação.

Nesse sentido, a arquitetura Visão de Proveniência Aberta - *Open Provenance Vision* [194] fornece diretrizes arquiteturais para dar suporte à interoperabilidade, por meio de vocabulário controlado de modelos de proveniência, formatos de serialização, tais como RDF e XML e APIs, permitindo expressividade quanto a proveniência de sistemas individuais.

Com a Visão de Proveniência Aberta, a proveniência de sistemas individuais ou componentes pode ser expressa, conectada e consultada. Dessa forma, para ser uniformemente consultável, Moreau [194] afirma que a proveniência deve ser representada usando descrições ontológicas do que aconteceu, de forma independente de tecnologia. Vários modelos estão se destacando na literatura, tais como Provenir [220], OPM [197] e PML [182], entre outros, para interoperabilidade de informações de proveniência, alinhados com essa visão de proveniência. Modelos de proveniência são abordados na Seção 4.6.

Ambas, a Web e a Web Semântica, são tecnologias a serem exploradas para representar, tornar acessível, consultar e permitir raciocínio lógico sobre proveniência [194]. Para expor informações na Web faz-se uso de URIs, um sistema de identificar recursos globalmente e protocolos, como HTTP para acessar recursos.

As tecnologias utilizadas envolvem RDF, permitindo recursos serem referenciados por URIs, onde sua estrutura de triplas facilita a representação gráfica. São utilizadas as Linguagens SPARQL [213] para consultas e OWL para definições ontológicas e raciocínio lógico. Moreau [194] descreve alguns exemplos de publicar proveniência na Web, conforme segue.

O sistema *Scientific Annotation Middleware* - SAM [199] é um sistema precursor no uso de tecnologias da Web Semântica. Oferece uma agenda eletrônica que captura proveniência de experimentos científicos. Adota a abordagem *Webdav* e identificadores URI, permitindo a navegação de informações de proveniência.

Zhao et al [253] identificam anotações em objetos e caminhos - *paths* de derivação, onde a Web é criada dinamicamente por meio de raciocínio ontológico, processamento de anotações e inserções de *links*. Descreve hipertextos gerados dinamicamente sobre documentos de proveniência, dados, serviços e *workflows*.

Zhao et al [255] visualizam informações de proveniência a partir de quatro diferentes níveis: i. organizacional, sobre quem rodou o *workflow*; ii. processo, como um *log*

de evento; iii. dados, capturando sua derivação; e iv. conhecimento, uma anotação sobre todos os anteriores. Em [254], são utilizadas as tecnologias: RDF para representar proveniência; especificação *Life Science Identifier* - LSIDs que é um esquema identificador único globalmente, para recursos de informação no domínio de ciências da vida, o qual é baseado em padrões da Internet e utiliza também ontologias para apresentar uma visão semântica comum.

O sistema Tupelo [54] fornece uma implementação do *Open Provenance Model* - OPM, contribuindo para ler e escrever informações OPM para um contexto de metadados RDF, assim como uma representação de alto-nível do OPM, a qual pode ser estendida para outras serializações OPM. Sua API fornece representações dos conceitos, juntamente com representações de arcos que os conectam em grafos OPM [200]. A próxima seção aborda modelos de proveniência, seguida de uma análise comparativa.

4.6 Modelos de Proveniência

A representação de proveniência necessita ser declarada para trazer evidências de transformações locais e derivações de modo coerente [196]. Nesse sentido, Modelos de Proveniência contribuem para interoperabilidade de informações de proveniência, pois constituem-se como um modelo comum para gerenciamento dessas informações. Seu uso aumenta o entendimento dos usuários sobre respostas geradas e facilita a aceitação dos resultados. Os principais e mais utilizados modelos de proveniência são descritos na sequência.

4.6.1 *Open Provenance Model* - OPM

O Grupo Incubador de Proveniência do W3C - PROV-XG [50] fornece um entendimento do estado da arte, na área de proveniência, para tecnologias da Web Semântica. Objetivando melhorar o entendimento mútuo das capacidades das linguagens de proveniência, este grupo propõe mapear muitas das linguagens de proveniência existentes tais como Provenir, PREMIS e a Linguagem de Marcação de Provas - *Proof Markup Language* (PML) para a Linguagem de Proveniência do OPM. Nesse sentido, OPM permite a troca de informações de proveniência.

A especificação OPM Versão 1.1 [196] descreve um modelo de proveniência como um grafo composto de nodos conectados através de arcos dirigidos. Um arco em um modelo OPM é uma dependência causal entre a fonte do arco (o efeito) e o destino do arco (a causa). Um nodo apresenta a forma oval e representa um Artefato. Uma caixa representa um Processo e um octógno representa um Agente. Os três conceitos principais de OPM são:

- Um Artefato é uma parte imutável de estado, que pode ter uma personificação em um objeto físico ou uma representação digital em um sistema computacional.
- Um Processo é uma ação ou séries de ações feitas ou causadas por artefatos, resultando em novos artefatos.
- Um Agente é uma entidade contextual agindo como um catalisador de um processo, gerenciando sua execução.

OPM também inclui os conceitos de *Accounts* e *Roles*. *Accounts* são identificados por um identificador único e representam uma descrição, em algum nível de detalhe, fornecido

por um ou mais observadores. Duas *Accounts* são iguais se e somente se elas têm o mesmo identificador. *Roles* são usados nas arestas *Used*, *WasGeneratedBy*, e *WasControlledBy*. O significado de um *role* é definido pela semântica do processo ao qual se refere.

Um grafo de proveniência é definido como um registro de execuções passadas (ou atuais), não sendo uma descrição do que pode ocorrer no futuro. OPM é um modelo de artefatos no passado, explicando como eles foram derivados, os quais podem ser no passado ou estar ainda em execução. Ou seja, OPM não pretende descrever o estado de artefatos e atividades de processos futuros [197].

Um grafo de proveniência objetiva capturar dependências causais entre entidades. Nodos podem ser artefatos, processos ou agentes, sendo conectados por arestas diretas que pertencem a uma das categorias do modelo. Uma aresta representa uma dependência causal entre sua fonte, a qual denota o efeito e seu destino, denotando a causa, expressando as seguintes dependências: um artefato foi gerado por um processo; um processo usou um artefato; um processo foi controlado por um agente; um artefato foi derivado a partir de outro artefato; e um processo foi disparado por outro processo (Figura 4.5).

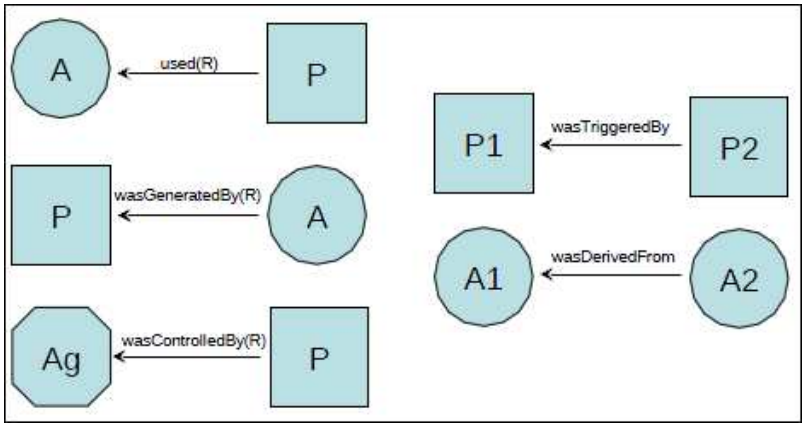


Figura 4.5: Nodos e arestas OPM [194].

OPM é um modelo teórico e possui um conjunto de regras de inferência, permitindo *reasoning* sobre dependências causais. O modelo é independente de tempo, mas é permitido anotar informação de tempo em um grafo de proveniência.

Freitas et al [131] investigam características e requisitos de proveniência na Web, descrevendo como o OPM pode ser usado como uma fundamentação para criação do W3P, um modelo de proveniência e uma ontologia, projetados para assegurar requisitos principais para a Web.

4.6.2 *Provenir*

Provenir é citada como uma Ontologia de Nível Superior - *Upper-level Ontology* [220, 219], a qual é utilizada para gerenciar proveniência em *e-Science* e codificada em *Ontology Web Language* - OWL-DL. Apresenta como conceitos principais: Agente, Processo e Dado. Como subclasse de Dados inclui Coleção de Dados e Parâmetros Espacial, Domínio e Temporal. É constituída por oito classes e onze propriedades, incluindo a Ontologia de Relação.

Provenir é um modelo expressivo quanto aos conceitos e relacionamentos modelados, rotulados como bem-definidos. Pode ser estendido para modelagem de informações de

proveniência complexas e específicas de domínio, habilitando análises em SWRL e W3C *Rule Interchange Format* - RIF. Esta ontologia tem aplicabilidade nas áreas biomédica e oceanografia em projetos reais de *e-Science*: Ontologia de Experimentos de Parasitas, modelando proveniência em pesquisa de parasitas [219] (Figura 4.6) e *Trident Ontology* - modelando proveniência no projeto oceanográfico Neptune.

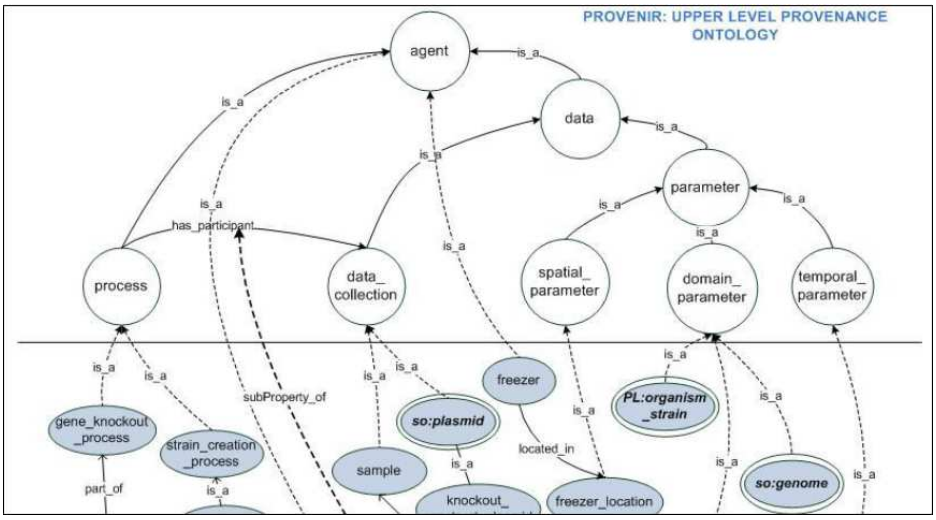


Figura 4.6: Ontologia de domínio e Provenir [220].

Comparando com OPM, um esforço similar para criar um modelo comum para representação de proveniência, Sahoo e Sheth [220] citam que esta Ontologia de Nível Superior é mais expressiva quanto a conceitos modelados e relacionamentos bem-definidos, facilitando sua extensão para modelagem de informações de proveniência complexas e específicas de domínio, tarefa que é difícil ou não possível usando OPM. Também, em OPM, a inferência é limitada, devido a sua estrutura de grafo genérica [220].

Sahoo e Sheth [220] afirmam que informações específicas do domínio são aspectos importantes de proveniência, principalmente em *e-Science*. Usar uma simples ontologia de proveniência monolítica para modelar detalhes a partir de diferentes domínios não é viável. Dessa forma, a ontologia de nível superior Provenir envolve o uso integrado de múltiplas ontologias, cada uma modelando metadados de proveniência, específicos a um domínio particular. Portanto, o uso deste modelo de proveniência como uma ontologia de nível superior facilita interoperabilidade acerca de ontologias de proveniência específicas de domínio.

4.6.3 *Proof Markup Language* - PML

PML baseia-se na Teoria da Provas, constituindo uma linguagem comum para representar e compartilhar explicações geradas por sistemas inteligentes. Descreve justificativas como uma sequência de manipulações de informação para gerar uma resposta, sendo referida como um prova. Apresenta uma abordagem modular, onde módulos podem ser utilizados individualmente, seja para Proveniência (PML-P), Justificativas (PML-J) ou assegurar confiança nos dados (PML-T) [182]. Conceitos e relações são especificados em OWL, facilitando reuso e extensão dessas ontologias. O domínio de aplicabilidade de PML é trilhar e gerenciar proveniência em *workflows* [185].

PML fornece vocabulário para justificativas de metadados cujo foco está em primitivas representacionais usadas para descrever propriedades de objetos identificados, tais como informação, linguagem e recursos (incluindo organização, pessoa, agente e serviços). Essas primitivas são extensíveis, sendo usadas para anotar a fonte da informação, como para representar fontes usadas e quem codificou a informação. Enquanto alguns termos tais como *InferenceRule* e *InferenceStep* são usados frequentemente por provadores de teoremas lógicos, são aplicados em qualquer configuração onde alguma inferência é usada para manipular informação [185]. Os três módulos PML são apresentados na sequência.

- *Provenance Ontology* - PML-P

  Em PML-P, uma instância de *IdentifiedThing* se refere a uma entidade do mundo real e suas propriedades. Inclui as subclasses *Information*, *Language*, *Source*, *SourceUsage* e *InferenceRule*. *Information* dá suporte a referências a informações sobre vários níveis de granularidade e estrutura. *Language* representa a linguagem em que a conclusão é representada. *Source* é extensível e refere-se a um recipiente de informações, como um Documento. *SourceUsage* é usado para associar *Information* e *Source*, como a data/hora de uso de uma fonte. *InferenceRule* permite codificar a regra que gerou a conclusão (a operação). Um *pmlp:Source* pode ser um documento, um agente, uma página Web, entre outros. PML-P fornece uma simples, mas extensível, taxonomia de fontes [111].

- *Justification Ontology* - PML-J

  PML-J representa as conclusões, zero ou mais conjuntos de antecedentes da conclusão e as medidas utilizadas para manipular informações, para obter conclusões, a partir do conjunto de antecedentes. O vocabulário para justificativas de dados concentra-se em primitivas de representação que explicam dependências entre objetos, representando como as conclusões são derivadas. Apresenta os conceitos *NodeSet* e *InferenceStep*. O *NodeSet* representa uma conclusão e um conjunto de passos alternativos, cada um dos quais pode ter uma justificativa alternativa para uma conclusão. Um *InferenceStep* representa uma justificativa para a conclusão do *NodeSet* e refere-se a um passo lógico de inferência. Um *InferenceStep* representa os detalhes, tais como *InferenceEngine*, *InferenceRule* e o conjunto de *NodeSets* antecedentes de uma justificativa para a conclusão do *NodeSet*. Esses termos podem ser mapeados para termos de *workflows* mais familiares. Da Silva et al [111] comentam que conclusões podem se referir a dados intermediários e antecedentes podem se referir a entradas de algum passo de processamento.

- *Trust Ontology* - PML-T

  A Ontologia *Trust Ontology* é utilizada para assegurar confiança nos dados. PML-T dá suporte a anotações de relações de confiança complexas, em conceitos de proveniência e justificativas.

A Figura 4.7 apresenta um exemplo de uso de PML para proveniência na interface de usuário, onde a proveniência da consulta é estendida para capturar usos subsequentes do resultado de consulta [173].

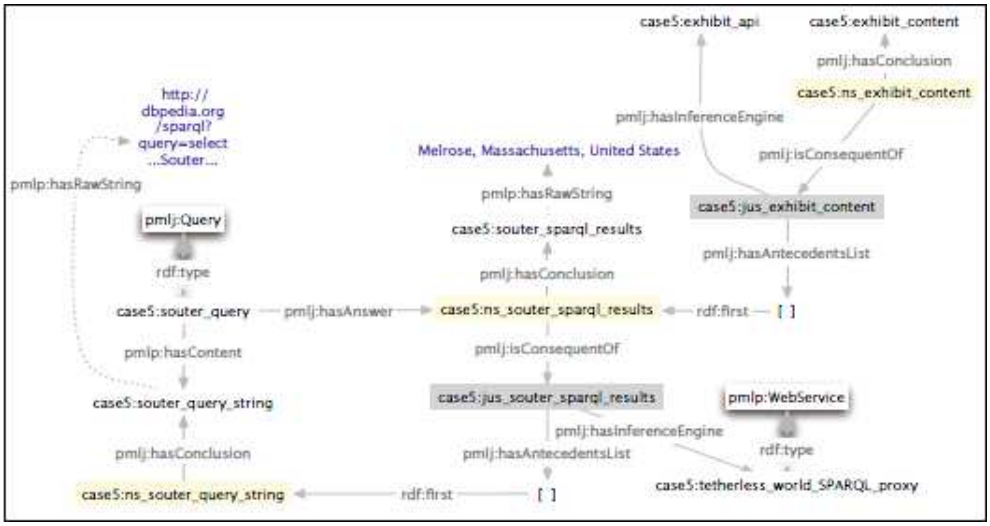


Figura 4.7: Consulta usando PML [173].

4.7 Análise Comparativa dos Modelos de Proveniência

A Figura 4.8 mostra os Esquemas dos Modelos de Proveniência, sendo: a) OPM, b) Provenir e c)PML. O Grupo Incubador de Proveniência do W3C - PROV-XG [50] apresenta um Relatório Final de Proveniência contendo uma análise de vocabulários existentes, incluindo mapeamentos acerca de termos em diferentes vocabulários [133].

Em [186] é feito um estudo comparativo dos modelos OPM e PML. A Tabela 4.1 apresenta conceitos baseados nas Versões de PML 2.0 e OPM 1.01, segundo o *Third Provenance Challenge* - PC3[1]. O mapeamento é definido em relação ao modelo OPM, o qual é estendido para incluir a ontologia Provenir: (i) entidades, (ii) relações entre entidades, e (iii) anotações em relações. Conceitos PML são usados a partir de suas duas ontologias, *Justification* (prefixo pmlj) e *Provenance* (prefixo pmlp). Destaca-se que OPM apresenta mínimo viés de codificação e PML se compromete com motores de inferência e representações em lógica.

A Tabela 4.2 apresenta um comparativo entre os modelos envolvendo critérios gerais e as Tabelas 4.3 e 4.4 envolvem critérios específicos de proveniência.

[1]URL:http://twiki.ipaw.info/bin/view/Challenge/ThirdProvenanceChallenge. Acesso em Fev/2014.

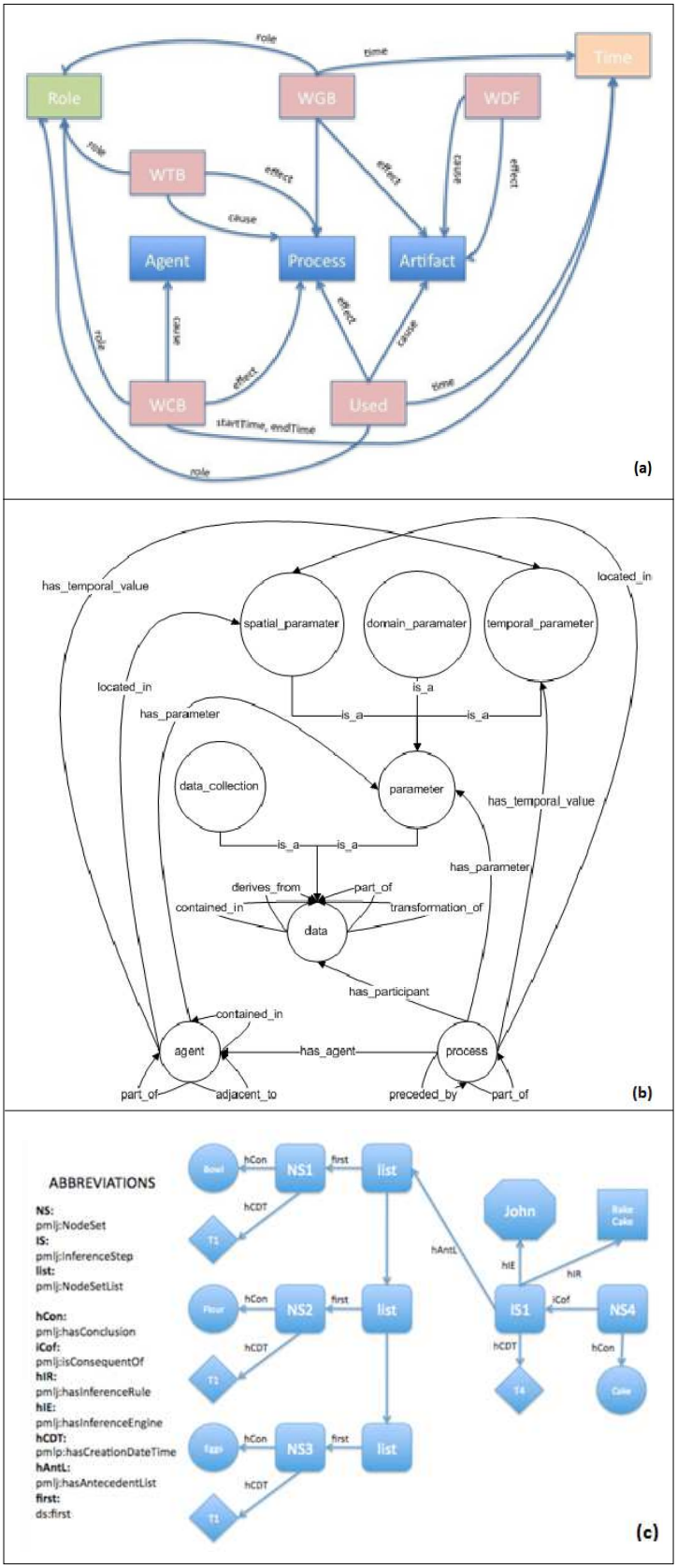


Figura 4.8: Esquemas dos modelos de proveniência a) OPM [196], b) Provenir [220] e c) PML [182].

Tabela 4.1: Mapeamento dos modelos de proveniência. Adaptada de [186].

Entidade OPM	**Equivalente PML**	**Equivalente Provenir**
Artifact	pmlj:Nodeset (+ pmlp:Information, pmlp:Source)	Data
Process	pmlj:InferenceStep (+ pmlp:InferenceRule)	Process
Agent	pmlp:InferenceEngine	Agent
Relação OPM	**Equivalente PML**	**Equivalente Provenir**
Used	pmlj:hasAntecedentList (+ pmlj:NodeSetList)	has_participant
WasGeneratedBy	pmlj:isConsequentOf (+ pmlp:hasConclusion)	has_participant
WasControlledBy	pmlp:hasInferenceEngine (+ pmlp:hasSourceUsage)	has_agent
WasDerivedFrom	pmlp:hasSourceUsage	derives_from
WasTriggeredBy	-	preceded_by
Overlaps	-	-
OPM Anotações em Relações	**Equivalente PML**	**Equivalente Provenir**
OTime	pmlp:hasCreationDateTime (+ xmls:DateTime)	Temporal Parameter
Role	-	-
Account	-	-

Vários quesitos podem ser levados em consideração quando do uso de um Modelo de Proveniência, conforme abordado na sequência.

- A Ontologia Provenir foi desenvolvida em OWL-DL, sendo classificada como Ontologia de Nível Superior - *Upper-level Ontology*. Esta ontologia possui operadores de consulta bem-definidos para proveniência, apresentando aplicabilidade em cenários reais.

- PML 2 tem amplo intervalo de conceitos e relacionamentos não contemplados por OPM e Provenir. Essas incluem propriedades específicas descrevendo e estruturando artefatos, agentes/fontes, processos e passos de inferência. Entretanto este modelo não especifica os relacionamentos *WasDerivedFrom*, *WasTriggeredBy* e *Overlaps* de OPM. Esse modelo não permite anotações adicionais.

- OPM permite a geração de proveniência independentemente de tecnologia. A Ontologia OPMO permite expressividade de conceitos e relações. Esta ontologia é proposta pelo W3C como um modelo comum para geração de proveniência, sugerindo um mapeamento dos demais modelos de proveniência para OPM. Considerando que OPM é adequado para o contexto desta tese, o qual pode ser estendido para gerar proveniência quanto a correção de tendências em séries temporais, assim como devido às descrições citadas, o mesmo é o modelo escolhido para reuso e extensão quanto a artefatos, processos e agentes envolvidos em um processo de *detrending*.

Como o objetivo dessa tese é a definição de um modelo proveniência na extração de tendências em séries temporais, o próximo capítulo aborda sobre os materiais e métodos relacionados ao seu desenvolvimento.

Tabela 4.2: Modelos de Proveniência - Critérios Gerais. Fonte: Os Autores.

Modelo de Proveniência/Critério	OPM	PROVENIR	PML
Descrição	OPM é um modelo abstrato para explicar como artefatos foram derivados. É independente de tecnologia para propósitos de interoperabilidade. Utiliza um grafo baseado em um conjunto de regras sintáticas e restrições topológicas.	PROVENIR é uma ontologia para gerenciamento de proveniência em *e-Science*.	PML é uma linguagem comum para representar e compartilhar explicações geradas por vários sistemas inteligentes. Usa três ontologias para codificar informações de respostas de agentes.
Propósito	*Workflow*	Manipulação da informação/*Workflow*	*Workflow*
Versão	V.1.01	V.1	V.2
Órgão/IE Responsável	Pesquisadores envolvidos com *Provenance Challenge Series*.	Kno.e.sis Center, Wright State University, USA	*Inference Web Group*
Conceitos e três sub-níveis superiores	Artifact, Process e Agent	Agent - Process e Data Data - Data_Collection e Parameter; Parameter - spatial_parameter, domain_parameter e temporal_parameter	list - pmlj:NodeSetList IS - pmlj: InferenceStep NS - pmlj: NodeSet
Dimensões	3 classes principais	8 classes, 11 propriedades (Relation Ontology - RO)	3 módulos
Modularização	OPM - Especificação versão 1.1 (modelo abstrato); OPMV - Vocabulário do OPM; OPMO - Ontologia do OPMX - Schema XML do OPM; OPM4J - biblioteca Java do OPM	–	Dividida em 3 módulos para reduzir a manutenção e diminuir custos: *Provenance Ontology* (PML-P); *Justification Ontology* (PML-J) e *Trust Ontology* (PML-T)
Uso de Padrões de Modelagem de Ontologias	–	–	–
Vantagens	Foco em *workflows*, define um conjunto de conceitos principais para entidades gerais (artefatos, agentes e processos) e relações em *workflows* (WasGeneratedBy e WasControlledBy).	Modelo comum para representação de proveniência. Expressivo quanto a conceitos modelados e relacionamentos nomeados bem-definidos, podendo ser estendido para modelagem de informações complexas e específicas de domínio; possibilita análises em SWRL e RIF.	Ontologia adicionada para W3Cs OWL; Descreve justificativas como uma sequência de manipulações de informações usadas para gerar uma resposta, referidas como uma prova (*proof*).
Desvantagens	Profile OWL está em evolução para adaptar a especificação OPM.	Ausência de padrões de modelagem de ontologias.	Comparando com OPM, não especifica as relações *WasDerivedFrom*, *WasTriggeredBy* e *Overlaps*, assim como propriedades usadas para linkar *Nodesets*, *InferenceSteps* e *Agents*; não permite anotações adicionais.

Tabela 4.3: Modelos de Proveniência - Critérios Específicos. Fonte: Os Autores.

Modelo de Proveniência/Critério	OPM	PROVENIR	PML
Classificação Modelo de Proveniência	Modelo de Proveniência	*Upper-Level Ontology*	Linguagem de Marcação de Provas
Distribuição	gratuita	gratuita	gratuita
Linguagem	Existem duas representações do grafo OPM: 1. Mapeamento para RDF, segundo a ontologia OPMO, sendo representado em notação N3 e 2. Serialização em XML compatível com o esquema OPMX.	OWL-DL	KIF e a especificação PML é sincronizada com sua ontologia OWL.
Formalização	Consiste de uma definição *set-theoretic* do modelo de dados, uma definição das inferências pelos fechamentos transitivos permitidos, descrição formal de como o modelo pode ser usado para expressar dependências em computações passadas e uma descrição do tipo de inferências baseadas em tempo que são suportadas.	Lógica ALCH	Usa *Proof Theoretic Foundation* para seu modelo mas tem sido usada para modelar proveniência em componentes menos formais e estatísticos.
Construtos baseados em	Conceitos de *workflow*	Ontologia de Fundamentação	Teoria de Provas
Baseada em	*Provenance Challenges*	*Open Biomedical Ontologies*(OBO)	*Inference Web Data*
Conceitos de proveniência	Artefatos, Processos e Agentes.	Baseado em dois conceitos primitivos da ontologia filosófica "*occurrent*" e "*continuant*", são definidos três classes básicas: Data, Agent (especializações da classe *Continuant*) e Processos (sinônimo da classe *Occurrent*).	Artefatos: pmlj: Nodeset e pmlp: Information; Processos: pmlj: InferenceStep e pmlp:InferenceRule Agentes: pmlp: InferenceEngine
Dimensão Espacial	-	Classe Parameter e subclasse spatial_parameter	-
Dimensão Temporal	Tempo: pode estar associado com ocorrências instantâneas em um processo (OTime).	Classe Parameter e subclasse temporal_parameter	pmlp: hasCreationDateTime e xmls: DateTime
Dimensão Temática (específicas do domínio)	Account (relação de anotação) e opmGraph	Classe Parameter e subclasse domain_parameter	-

Tabela 4.4: Modelos de Proveniência - Critérios Específicos (Cont.)

Modelo de Proveniência/Critério	OPM	PROVENIR	PML
Aplicabilidade em diferentes domínios	Biologia	Áreas BioMédica e Oceanografia (projetos reais de *e-Science*)	Sistemas inteligentes (sistemas de respostas de questões Web híbridos, texto analítico, provadores de teoremas, processadores de tarefas, *Web Services*, *rule engines* e componentes de aprendizagem de máquina).
Inferência	Sim, oferece inferência pelo *reasoner* em classes, propriedades e verificação de integridade. Apresenta 3 tipos de inferências: 1. completa (*WasTriggeredBy* e *WasDerivedFrom*); 2. múltiplos passos (causas indiretas de derivação) e; 3. expansão de *profile*.	Sim	Sim, oferece inferência pelo *reasoner* em classes, propriedades e verificação de integridade.
Uso de vocabulários/Esquemas	OPMV	-	Especificação XML *Schema* para tipos primitivos
Consulta de proveniência	Apesar de OPM não especificar protocolos específicos para consultar repositórios de proveniência, as formas *where*, *why*, *how* e *route* são investigadas em linguagens de consulta/atualização, consideradas como consultas sob o grafo de proveniência.	Utiliza-se o motor de consulta de proveniência *Oracle RDF Store*, sendo propostos operadores de consultas: *provenance()*, *provenance_context()*, *provenance_compare()* e *provenance_merge()*.	Tradução de consultas por agentes em representação interna usando *Knowledge Interchange Format* (KIF)
Aspectos positivos quanto a geração de informações de proveniência	Possibilita geração de proveniência de maneira independente de tecnologia para *workflows*; OPMO permite completa expressividade e inferências; modelo proposto como consenso comum por pesquisadores; proposta de mapeamento pela Incubadora de Proveniência W3C dos modelos de proveniência existentes para o OPM.	Operadores para consultas de proveniência; aplicabilidade em aplicações reais.	Possibilita geração de proveniência independente do domínio em *workflows*, sendo uma interlíngua projetada para suportar interfaces de respostas de questões em sistemas inteligentes; PML é uma ontologia adicionada para W3Cs OWL.
Aspectos negativos quanto a geração de informações de proveniência	Em OPM, a inferência é limitada devido a estrutura de grafo genérica, assim como inferências transitivas no grafo de proveniência ainda não são capturadas por OWL.	-	Inferência transitiva não tem sido capturada em OWL.

CAPÍTULO 5

MATERIAIS E MÉTODOS

Este capítulo descreve os materiais e métodos utilizados para a definição do Modelo de Proveniência para enriquecer semanticamente o passo de extração de tendências da análise de séries temporais. As metodologias utilizadas para o desenvolvimento e avaliação das ontologias são descritas, assim como os artefatos computacionais utilizados.

5.1 Metodologias

As metodologias *Ontology Development* 101 [204] e *NeOn Methodology* [233] foram utilizadas de forma complementar para o desenvolvimento das Ontologias de Domínio (*Time Series Ontology* e *Detrend Ontology*) e o Modelo de Proveniência (*Detrend Provenance Model*). O uso complementar se justifica porque a metodologia *NeOn* [233] permite a escolha de cenários e ciclos de vida alternativos para modelagem, incluindo cenários específicos para modelar questões relacionadas ao reuso de ontologias e de declarações semânticas, apresentando passos mais detalhados para o desenvolvimento das ontologias, assim como contempla o uso de um projeto modular. Ambas as metodologias são descritas na sequência.

5.1.1 Metodologia *Ontology Development* 101

Noy e McGuinness [204] estabeleceram regras quanto às decisões de projeto para o desenvolvimento de ontologias:

- Não há uma maneira correta de modelar um domínio, ou seja, depende da aplicação desejada.
- O processo de desenvolvimento de uma ontologia é iterativo, o qual continuará no ciclo de vida da ontologia.
- Conceitos da ontologia devem estar próximos a objetos (nomes) e relacionamentos (verbos) no domínio de interesse.

Os passos metodológicos da Metodologia *Ontology Development* 101 para o desenvolvimento das ontologias incluem:

- Determinar o domínio e escopo da ontologia.
- Identificar, a partir do domínio, um conjunto de questões de competência que a ontologia deve responder.
- Considerar reuso.
- Enumerar itens relevantes.
- Definir classes e hierarquias de classes.

- Definir propriedades e restrições.
- Definir instâncias.

Quanto à definição da Ontologia TSO, as questões de competência referentes à proveniência de séries temporais foram definidas segundo o Modelo conceitual W7 [215], contribuindo para definir, capturar e usar dados de proveniência, apresentando sete elementos inter-conectados: *What?, When?, Where?, How?, Who?, Which? e Why?*. Esses elementos podem ser usados para trilhar eventos que afetam os dados durante seu tempo de vida. Este modelo de proveniência é geral e extensível para capturar a semântica de proveniência para dados em diferentes domínios [215].

- Proveniência de algum dado (D) é um conjunto de n-tuplas:

P(D) = 1. *What*, 2. *When*, 3. *Where*, 4. *How*, 5. *Which*, 6. *Who*, 7. *Why*

- Conjunto identificado para Ontologia de Séries Temporais (TSO):

P(TSO) = 1. *What*, 2. *When*, 3. *Where*, 4. *How*, 5. *Which*

Definições dos elementos conceituais do Modelo W7:

1. *What?* - um evento (mudança de estado) que afetou as séries temporais
   Evento(s): Geração, Decomposição e Modelagem das séries temporais
2. *When?* - refere-se ao tempo em que o evento ocorreu
   Evento: Geração das séries temporais
3. *Where?* - localização do evento
   Evento: Localização do armazenamento das séries temporais
4. *How?* - ação que conduziu ao evento
   Ação: Observação das séries temporais, Inserção de séries temporais
   Eventos: Classificação das séries temporais, Geração das séries temporais, Armazenamento das séries temporais.
5. *Which?* - usuários, programas ou instrumentos utilizados no evento
   Evento: Geração das séries temporais

- Conjunto identificado para análise do componente tendência das séries temporais (T):

P(T) = 1. *What*.

Eventos: Geração das séries temporais e Modelagem das séries temporais.

Os passos da Metodologia *Ontology Development* 101 correspondem, de forma geral, ao Cenário 1 e, especificamente quanto ao reuso, estes correspondem aos Cenários 2 a 6 da Metodologia *NeOn*, a qual é descrita na sequência.

5.1.2 Metodologia *NeOn*

No contexto da Engenharia de Ontologias, a Metodologia *NeOn* [233] se destaca por apresentar uma variedade de formas alternativas para o desenvolvimento de ontologias. Essa metodologia considera o desenvolvimento de ontologias como um processo centrado no reuso, dado o número de ontologias online atualmente disponíveis. Essa metodologia permite o desenvolvimento de uma rede de ontologias (*ontology network*), onde diferentes recursos podem ser gerenciados por diferentes pessoas (especialistas do domínio, desenvolvedores de ontologias, entre outros) e em diferentes organizações.

A metodologia *NeOn* pode ser usada para o desenvolvimento de ontologias (ou uma rede de ontologias), apresentando nove cenários flexíveis que descrevem situações comuns de desenvolvimento colaborativo, dando especial ênfase no reuso e na re-engenharia de recursos de conhecimento, ontológicos e não-ontológicos.

Essa metodologia também pode ser usada com a Iniciativa *Linked Open Data* [75], pois é baseada em recursos de conhecimento reutilizados e re-engenharia, assim como em recursos de mapeamento. Publicar dados *linkados* é um processo que envolve várias atividades, decisões de projeto e o uso de tecnologias. Entre as atividades, se destacam [233]: identificação das fontes de dados; modelagem do vocabulário, onde ontologias para modelar os dados contidos nas fontes selecionadas são desenvolvidas; geração de dados RDF; publicação de dados RDF; e *linking* dos dados RDF com outros conjuntos de dados da Web Semântica. A mais importante recomendação é reutilizar, tanto quanto possível, recursos disponíveis que modelam o conhecimento necessário.

A Figura 5.1 apresenta o conjunto dos nove cenários para o desenvolvimento de ontologias e redes de ontologias[1]. Os nove cenários da Metodologia *NeOn* são brevemente descritos na sequência.

- Cenário 1. A partir da especificação para a implementação. Nesse caso a ontologia é desenvolvida a partir do zero, sem reutilizar recursos de conhecimento disponíveis.
- Cenário 2. Reuso e re-engenharia de recursos não-ontológicos.
- Cenário 3. Reuso de recursos ontológicos, incluindo ontologias como um todo, módulos de ontologias e/ou declarações de ontologias na forma de triplas (*subject - predicate - object*).
- Cenário 4. Reuso e re-engenharia de recursos ontológicos.
- Cenário 5. Reuso e fusão (*merging*) de recursos ontológicos.
- Cenário 6. Reuso, fusão e re-engenharia de recursos ontológicos.
- Cenário 7. Reuso de padrões de projeto de ontologia (*Ontology Design Patterns* - ODPs).
- Cenário 8. Re-estruturação de recursos ontológicos, envolvendo modularização, cortes, extensões e/ou especialização de recursos a serem integrados na ontologia sendo desenvolvida.

[1]Considerando que a metodologia *NeOn* pode ser utilizada para o desenvolvimento de ontologias assim como de redes de ontologias, o termo ontologia será utilizado deste ponto em diante.

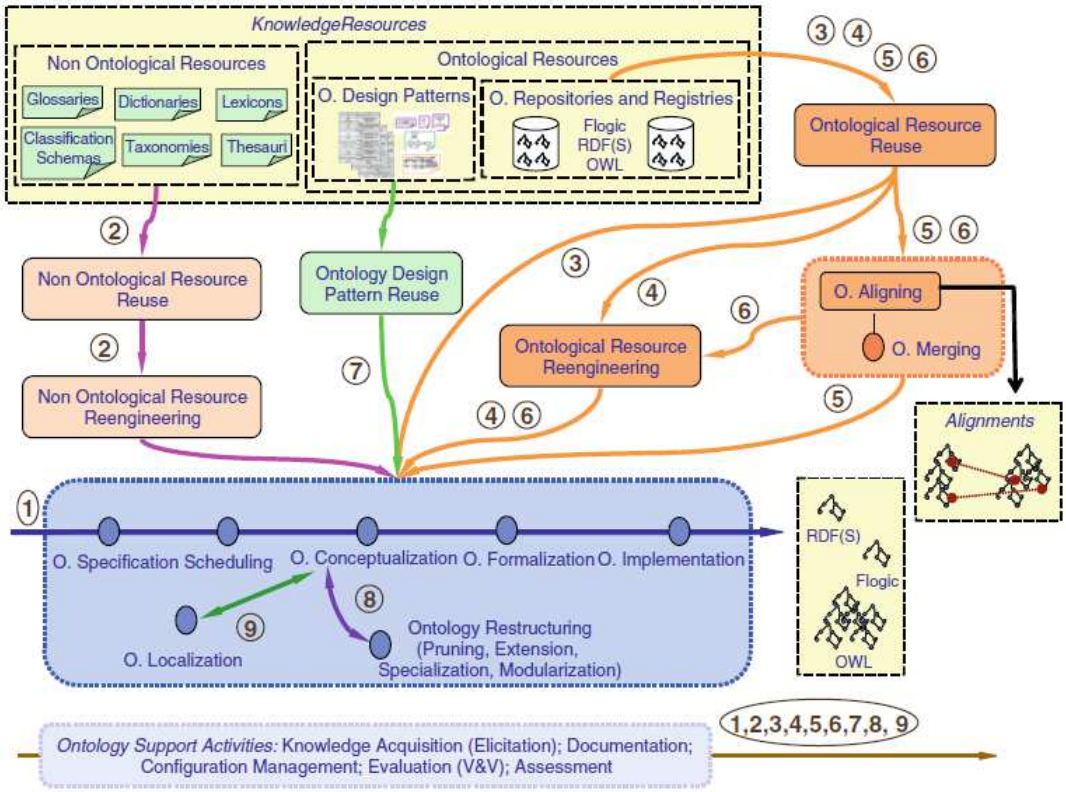


Figura 5.1: Cenários da metodologia *NeOn* [233].

- Cenário 9. Localização de recursos ontológicos, incluindo a adaptação para outros idiomas e comunidades culturais, produzindo ontologias multi-linguais.

A aquisição de conhecimento, documentação, gerenciamento de configuração, avaliação, entre outros, devem ser realizados durante o desenvolvimento inteiro da ontologia, ou seja, em qualquer cenário relacionado. Os cenários também podem ser combinados em diferentes e flexíveis maneiras, mas qualquer combinação destes deve incluir o Cenário 1, visto que este contém as principais atividades para qualquer desenvolvimento, conforme pode ser observado na Figura 5.1, onde resultados dos demais cenários são integrados ao Cenário 1.

Para a definição das Ontologias, foram selecionados os seguintes cenários: 1 (base para desenvolvimento), 3 e 8, os quais são descritos na sequência, conforme [233]. A descrição dos demais cenários pode ser consultada na devida fonte, em [233].

5.1.2.1 Cenários Utilizados para Definição das Ontologias

Cenário 1. Primeiramente é feita a atividade de especificação dos requisitos da ontologia. Como resultado, obtém-se um Documento de Especificação dos Requisitos da Ontologia - *Ontology Requirements Specification Document* (ORSD) incluindo o propósito, o escopo, a linguagem de implementação, o grupo-alvo, os usos pretendidos para a ontologia, assim como o conjunto de requisitos que a ontologia deve satisfazer, na forma de Questões de Competência e um pré-glossário de termos.

Após a especificação do ORSD, é recomendado a pesquisa por recursos de conhecimento candidatos a serem reusados no desenvolvimento. Em seguida é feita a atividade de

escalonamento (*scheduling*), usando o ORSD e os resultados da pesquisa por recursos. Nessa atividade, são estabelecidos o ciclo de vida da ontologia e os recursos humanos necessários.

Na sequência, são realizadas as seguintes atividades:

1. Conceitualização da ontologia, onde o conhecimento é organizado e estruturado em modelos significantes em nível do conhecimento.

2. Formalização da ontologia, onde o modelo conceitual é transformado em um modelo semi-computável.

3. Implementação da ontologia, onde um modelo computável é gerado em uma linguagem de implementação ontológica como OWL.

Como resultados deste Cenário, têm-se a ontologia implementada e vários documentos, como o ORSD, um documento de descrição da ontologia, um documento de avaliação da ontologia, gerados como saída de diferentes atividades.

Cenário 3. Reuso de Recursos Ontológicos. A Figura 5.1 descreve este cenário (linha 3), onde é feito o processo de reuso de recursos ontológicos, conforme as seguintes atividades:

1. Pesquisa de ontologias. Nessa atividade, desenvolvedores de ontologia pesquisam por recursos ontológicos que satisfaçam os requisitos em repositórios e registros tais como Swoogle [53], Watson [57] e Sindice [52].

2. Avaliação da ontologia. Nessa atividade, os recursos ontológicos obtidos na atividade anterior são inspecionados quanto ao conteúdo e granularidade, objetivando verificar se tais recursos satisfazem os requisitos de ORSD.

3. Comparação de ontologias. Os recursos ontológicos avaliados na atividade anterior são comparados conforme custos econômicos, clareza de código e qualidade de conteúdo.

4. Seleção de ontologia. Desenvolvedores devem selecionar o conjunto de recursos ontológicos que são mais apropriados, baseados nos critérios comparados na atividade anterior.

   Após definir os recursos ontológicos, é preciso definir o modo de reuso, podendo ser:

   - Os recursos ontológicos serão reutilizados como são.
   - A atividade de re-engenharia da ontologia deve ser aplicada aos recursos selecionados.
   - Alguns recursos serão mesclados para se obter um novo recurso ontológico.

   Antes de reutilizar tais recursos, independente de qualquer modo, é conveniente fazer a atividade de avaliação da ontologia.

5. Integração de ontologia. Desenvolvedores de ontologia devem incluir os recursos ontológicos selecionados por meio das atividades do Cenário 1.

Uma demonstração de reuso de recursos ontológicos [55] destaca que, dada a heterogeneidade de ontologias disponíveis online, considerando a perspectiva de qualidade, definir se uma declaração satisfaz as necessidades da ontologia sendo desenvolvida não é uma tarefa trivial. Na sequência, são listados alguns critérios para verificar o reuso das declarações:

- Verificar se as declarações da ontologia pertencem a uma ontologia com o mesmo ou similar escopo.
- Verificar o propósito das declarações da ontologia encontrada e o propósito da ontologia sendo desenvolvida para saber se estas são similares ou não.
- Verificar a clareza da informação na declaração, onde é necessário entender precisamente a informação codificada na declaração.
- Verificar o conteúdo da informação na declaração, onde a mesma deve ser informativa.
- Assegurar a corretude da declaração, a partir de uma perspectiva de modelagem, como o uso de uma relação sub-classe para modelar uma relação todo-parte (*part-of*).
- Verificar se a nomenclatura de entidades na declaração da ontologia reflete o significado pretendido da declaração conforme o contexto, ou seja, as declarações originais podem não fazer sentido em um outro contexto.

As declarações podem ser reutilizadas como se encontram, sem requerer mais esforços, entretanto, isso nem sempre acontece, conforme relata [233]. Em muitos casos, entidades criadas recentemente terão que ser renomeadas, onde são adicionadas restrições, alteradas quanto à hierarquia de classes, entre outras situações. Esse processo de re-engenharia do conhecimento reutilizado é necessário para se obter uma ontologia coerente, homogênea e bem-estruturada.

Como resultado deste Cenário, têm-se uma ontologia implementada, assim como um conjunto de documentos relacionados às diferentes atividades.

Cenário 8. Inclui os casos onde o conhecimento contido no modelo conceitual da ontologia deveria ser corrigido e reorganizado. Essa atividade é feita após a atividade de conceitualização da ontologia, envolvendo qualquer uma (e em qualquer ordem) das atividades que seguem.

- Atividade de modularização da ontologia. Desenvolvedores criam diferentes módulos de ontologias, facilitando o reuso do conhecimento.
- Atividade de corte da ontologia. Desenvolvedores excluem ramos da taxonomia considerados não-necessários.
- Enriquecimento da ontologia. Essa atividade pode ser feita por meio de qualquer uma das seguintes atividades:
  - Atividade de extensão da ontologia. Desenvolvedores estendem a ontologia, incluindo novos conceitos e relações.

– Atividade de especialização da ontologia. Desenvolvedores especializam ramos que requerem mais granularidade e incluem conceitos mais especializados e relações.

A re-estruturação da ontologia pode ser feita de modo independente ou como parte do processo de re-engenharia de recursos do Cenário 4. A principal saída deste Cenário é a ontologia que representa o domínio esperado.

5.1.2.2 Modelos de Ciclo de Vida da Ontologia

O processo de desenvolvimento de ontologias é visto como um caso específico de desenvolvimento de software. Um Modelo de Ciclo de Vida de Ontologia é definido como um modelo que descreve como desenvolver e manter um projeto de desenvolvimento de ontologia. Os modelos existentes são os Modelos de Ciclo de Vida de Ontologia *Waterfall* (Cascata) e Iterativo-Incremental [233].

No Modelo Cascata ocorre a representação dos estágios de uma ontologia, em fases sequenciais, como uma cascata. Um estágio concreto deve ser completado antes do início dos próximos estágios e nenhum retorno é permitido, exceto na fase de manutenção. A principal suposição é que os requisitos sejam completamente conhecidos, sem ambiguidades e considerados desde o início do desenvolvimento.

Este modelo é usado, entre outras situações, quando o projeto de ontologia cobre um domínio reduzido e bem-entendido. O conjunto de atividades de suporte inclui a aquisição de conhecimento no domínio em que a ontologia está sendo desenvolvida, a avaliação e a validação a partir das perspectivas do usuário de diferentes fases de saída, projeto, gerenciamento de configuração e documentação.

Existem cinco versões diferentes do modelo de ciclo de vida em cascata, as quais são feitas incrementalmente. Por exemplo, o modelo de quatro fases é base para o de cinco fases. Os modelos de quatro e cinco fases relacionados com esta pesquisa encontram-se descritos na sequência.

- **Modelo Cascata de Quatro Fases**

O Modelo Cascata de Quatro Fases representa os estágios de uma ontologia, contendo as seguintes fases, conforme a Figura 5.2.

- Fase inicial: nesta fase é produzido o ORSD, incluindo os requisitos que a ontologia deveria satisfazer e levando em consideração o conhecimento sobre o domínio concreto.

- Fase de projeto: a saída desta fase deve ser um modelo formal satisfazendo os requisitos da fase anterior. Esse modelo formal não pode ainda ser usado por computadores, mas pode ser reusado por outras ontologias.

- Fase de implementação: nesta fase, o modelo formal é implementado em uma linguagem ontológica. A saída é uma ontologia implementada em RDF(S), OWL ou outra linguagem que pode ser usada por aplicações semânticas ou por outras ontologias.

  As fases de projeto e implementação são normalmente feitas juntas quando se utiliza ferramentas de desenvolvimento de ontologias tais como Protégé [166], *NeOn Toolkit* [233], entre outras.

- Fase de manutenção: se durante o uso da ontologia, erros ou algum conhecimento esteja ausente, desenvolvedores podem retornar à fase de projeto. Adicionalmente, nesta fase, a geração de novas versões para a ontologia podem ser geradas.

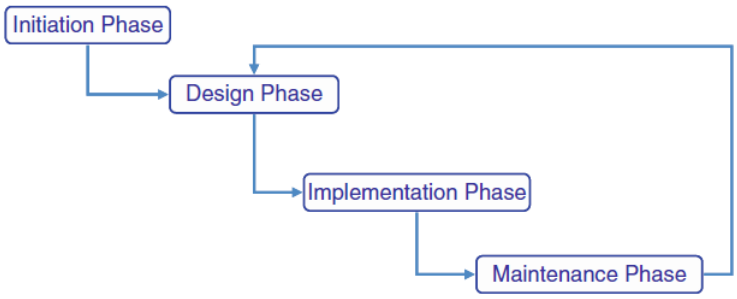


Figura 5.2: Modelo de ciclo de vida em cascata de quatro fases [233].

- **Modelo Cascata de Cinco Fases**

O Modelo Cascata de Cinco Fases estende o modelo de quatro fases, com a fase do reuso de recursos ontológicos. O propósito é obter um ou mais recursos ontológicos a serem reutilizados. A saída deve ser ou um modelo informal ou um formal usado na fase de projeto ou um modelo implementado em uma linguagem ontológica, a ser usado na fase de implementação.

Além do Modelo Cascata, existe o Modelo Iterativo-Incremental, onde o desenvolvimento é feito por um conjunto de iterações ou mini-projetos com uma duração fixa. Cada iteração é similar ao projeto de ontologia, usando qualquer tipo do Modelo em Cascata. Esse modelo é indicado para os casos em que a equipe de desenvolvimento envolve grande número de desenvolvedores, diferentes domínios e os requisitos não são completamente bem-entendidos. O ORSD nesse caso pode ser dividido em diferentes subconjuntos.

Os cenários e os modelos de ciclo de vida são intrinsecamente relacionados. Quanto aos Cenários 1, 3 e 8, identificados nesta tese, os mesmos são relacionados às respectivas Fases do Modelo Cascata. No Cenário 1, a ontologia é desenvolvida incluindo as atividades de especificação, conceitualização e implementação, relacionadas ao modelo quatro fases (inicial, projeto, implementação e manutenção). No Cenário 3, usado para desenvolver ontologias pelo reuso de recursos ontológicos, é representado pelo modelo de cinco fases, que inclui o reuso. E quanto ao Cenário 8, usado para desenvolver ontologias pela re-estruturação de recursos ontológicos, o mesmo é relacionado às atividades do Cenário 1, é relacionado pelo modelo de quatro fases.

5.1.2.3 Avaliação das Ontologias

Conforme a Metodologia *NeOn* [233], a avaliação de ontologias é uma atividade para verificar a qualidade técnica de uma ontologia. São apresentados dois diferentes tipos de avaliação de ontologia, Validação e Verificação. O primeiro tipo compara o significado das definições da ontologia e o modelo pretendido, correspondendo à resposta à questão: "*Vocês estão produzindo a ontologia certa?*". O segundo tipo compara a ontologia com o documento de especificação da mesma (requisitos e questões de competência), assegurando

que a ontologia está construída corretamente, respondendo à questão "*Vocês estão produzindo a ontologia do modo certo?*".

A avaliação de ontologias consiste de métodos de avaliação que podem ser classificados em três categorias principais:

- Avaliação Estrutural: considera a estrutura lógica da ontologia, onde a avaliação ocorre em níveis sintático e semântico e livre de contexto. Algumas medidas estruturais que podem ser usadas são: profundidade, largura, modularidade, entre outras. Em uma avaliação estrutural, a ontologia é considerada um objeto de informação.

- Avaliação Funcional: significa avaliar a ontologia em seu contexto de uso. Medidas funcionais incluem acordo de especialistas, verificação de tarefas e de tópicos. Nesse caso a ontologia é visualizada como um objeto de informação e sua conceitualização pretendida.

- Avaliação de Usabilidade: foca em quão bem as ontologias contemplam a conceitualização pretendida. Medidas são relacionadas a *profile* da ontologia (metadados da ontologia), ou seja, sobre a ontologia em si e seus elementos. Três níveis de usabilidade podem ser definidos: i. Reconhecimento sobre as anotações da ontologia; ii. Eficiência, envolvendo o estudo de quão eficiente a ontologia dá suporte às necessidades do usuário; e iii. Nível de interface, relacionado à conexão com interfaces de usuário baseadas em ontologias.

Suárez-Figueroa et al [233] explicam que existem muitas técnicas para todos os diferentes métodos de avaliação de ontologias e o desafio é como melhor selecionar os métodos adequados para o caso em questão e os dados e recursos disponíveis.

Quanto à avaliação do modelo definido, a modularização das ontologias contribuiu para a realização de avaliações individuais em cada módulo (ontologia). Nesta tese, a avaliação foi feita utilizando a dimensão da avaliação funcional, onde as ontologias são avaliadas em seu contexto de uso, contemplando as atividades de validação e verificação.

Para a escolha dos avaliadores, foram considerados os seguintes perfis: especialistas do domínio de séries temporais e ontologistas. A abrangência da escolha dos avaliadores é regional, visto que os avaliadores recebiam uma explicação sobre o domínio relacionado e sobre o desenvolvimento de ontologias.

Para a realização das avaliações, os avaliadores receberam os seguintes documentos:

- Termo de Compromisso, explicando sobre o que é a avaliação e sobre questões éticas.

- Lista de Questões de Competência, contendo questões que a ontologia deve ser capaz de responder.

- Questionário de Avaliação, onde o avaliador responde às questões relacionadas à avaliação da ontologia.

- Lista das referências utilizadas para definir e comentar as classes das ontologias.

- Questionário caracterizando o perfil do avaliador em termos da experiência na área.

- Documentação online da ontologia, onde o avaliador recebia os arquivos para abrir a documentação *online* em um navegador Web. Dessa forma, o avaliador pode acessar todos os elementos da ontologia, incluindo classes, relacionamentos e instâncias,

assim como visualizar comentários e rótulos, associação das instâncias com a DBpedia, entre outras informações. A documentação permitiu que os avaliadores tivessem acesso às ontologias sem a necessidade de instalar software específico de edição para visualizá-las.

Os documentos das avaliações encontram-se, respectivamente nos Apêndices A a D deste documento, referentes à documentação das ontologias TSO, DO e DPM e um questionário sobre o perfil do avaliador.

5.2 Artefatos Computacionais Utilizados para a Definição das Ontologias

Na sequência são descritas os artefatos utilizados para a definição das ontologias.

- **Ferramentas:**
  - Protégé 4.1 [166]: utilizada para o desenvolvimento da ontologia em OWL-DL 2.0 e a geração da base de conhecimento.
  - *Plugin Ontograf* 1.0 [20] permite visualizar relações dinâmicas de indivíduos e classes da ontologia.
- **Linguagens:**
  - *Web Ontology Language* (OWL 2.0) [151, 66]: essa linguagem ontológica é recomendada para uso pelo Consórcio W3C, a qual é baseada em Lógica Descritiva (OWL-DL).
  - SPARQL 1.1 [118]: permite o desenvolvimento de consultas na ontologia.
  - *Semantic Web Rule Language* (SWRL) [154]: utilizada para desenvolvimento de regras. O *reasoner* Pellet [227] permite trabalhar com regras *DL-Safe Rules*, as quais são aplicáveis somente a indivíduos nomeados da ontologia.
- **Raciocinador lógico:**
  - Pellet 2.2 [227]: o *reasoner* Pellet é gratuito, permite fazer inferências lógicas e verificar a consistência da ontologia.
- **Ferramentas de busca de Ontologias:**
  - *Swoogle - Semantic Web Search* [53]: segundo [117], possibilita efetuar pesquisas quanto a ontologias, documentos e termos. É um indexador e sistema de recuperação para a Web Semântica de documentos RDF ou OWL. Extrai metadados para cada documento recuperado, computando relações entre eles. Uma propriedade abordada é o *rank*, caracterizando uma medida de importância de um documento na Web Semântica. Sua arquitetura apresenta quatro componentes principais: descoberta de documentos da Web Semântica, criação de metadados, análise de dados e interface. A arquitetura é centralizada nos dados e extensível, onde componentes trabalham independentemente e interagem através de um banco de dados.

- *OntoSearch - Ontology Google* [22]: conforme [252], é um motor de busca de ontologias. Combina *Google Web APIs* com uma técnica de visualização hierárquica. Buscas podem ser feitas usando palavras-chave, em determinados tipos de arquivos de ontologia, permite visualmente inspecionar arquivos para verificar sua relevância. É baseado nas tecnologias Java, *Java Server Pages* - JSP, Jena e JBoss.
- *Plugin Watson* [56]: a partir da ferramenta *NeOn Toolkit* [233], este *plugin* possibilita a descoberta, inspeção e reuso de declarações de ontologias originárias, a partir de ontologias disponíveis na Web, apresentando as declarações selecionadas diretamente no ambiente de desenvolvimento da ontologia. Essa busca contribui, não só para o desenvolvimento de ontologias, mas para estender ou re-estruturar conceitos a partir de uma ontologia existente, colaborando não só para a atividade de reuso, quanto para um melhor entendimento do domínio, em relação às diferentes maneiras de modelagem.
- *Watson Web Interface* [57]: é uma versão da interface Watson, a qual contribui para a busca de ontologias na Web.

Destaca-se que tanto Swoogle quanto o *plugin* e a interface Watson Web foram utilizados para busca de ontologias a serem reutilizadas no desenvolvimento das ontologias propostas.

O próximo capítulo apresenta a definição do Modelo de Proveniência *Detrend*, desenvolvido com base nos materiais e métodos apresentados neste capítulo.

CAPÍTULO 6

DEFINIÇÃO DO MODELO DE PROVENIÊNCIA PARA EXTRAÇÃO DE TENDÊNCIAS EM SÉRIES TEMPORAIS

6.1 Introdução

Este capítulo apresenta a contribuição desta tese, a definição de um modelo de proveniência para a extração de tendências em séries temporais. Um projeto modular é considerado desde o início de sua definição, consistindo das seguintes ontologias (módulos): a primeira é relacionada à proveniência de séries temporais, a segunda descreve métodos que podem ser usados para extração de tendências (*detrending*) e, a terceira contém a combinação destas com a ontologia de um modelo de proveniência definido para reuso pelo Consórcio W3C, o qual é reutilizado e estendido para permitir interoperabilidade semântica. O modelo resultante permite a geração de informações de proveniência no passo de extração de tendências em séries temporais.

As ontologias apresentadas correspondem à *Time Series Ontology* (prefixo tso:), *Detrend Ontology* (prefixo do:) e *Detrend Provenance Model* (prefixo dpm:). Para escolha dos prefixos (*namespaces*) das ontologias, foi feito uma verificação em (prefix.cc), um localizador de *namespaces* para desenvolvedores RDF, sendo verificado que não existe até o momento os prefixos (tso, do e dpm) no banco de dados disponível para consultas, assim como nas ontologias disponíveis pesquisadas.

Nesse documento, os elementos das ontologias são apresentados entre parêntesis e o reuso, a partir de Ontologias da Web Semântica para Ciências da Terra e Ambientais (*Semantic Web for Earth and Environmental Ontology* - SWEET), entre outras, contém prefixos de suas ontologias. Todas as classes, propriedades de dados e de objetos são anotadas por meio da *tag* (rdfs:comment), identificando qual é a fonte da definição, assim como por meio da *tag* (rdfs:label) que é usada para rotular os elementos. Isto contribui para o entendimento dos conceitos, assim como permite saber qual a proveniência das definições. Quando aplicável, as instâncias das ontologias são associadas com a respectiva URL da base de dados semântica DBpedia [47], permitido interoperabilidade.

Este capítulo é dividido em três seções principais. A primeira descreve a definição da ontologia de proveniência em séries temporais. A segunda descreve os métodos estatítiscos usados para extração de tendências e a terceira é referente ao modelo de proveniência.

6.2 *Time Series Ontology* (TSO)

Dentre as formas para geração de proveniência, uma abordagem considera a geração de informações semânticas baseadas em ontologias, modelando conceitos e relacionamentos usados na geração de proveniência e contribuindo para inferências que descobrem conhecimento implícito, por meio de linguagens tais como *Resource Description Framework* (RDF) e *Ontology Web Language* (OWL). Ontologias fornecem as vantagens de descrição semântica do contexto, melhoramentos em consultas e provas de origem e busca de interoperabilidade dos dados gerados [196].

Na análise de séries temporais, informações de proveniência, tais como "Qual o tipo de observação das séries temporais?", "Como as séries temporais foram geradas?", "Qual

é o modelo de decomposição usado?", "Quais suposições foram consideradas?", "Como as séries temporais podem ser classificadas conforme as suposições?", "Qual o tipo de tendência considerada?", entre outras, permitem ao pesquisador interpretar melhor os dados e usar métodos estatísticos apropriados, especificamente desenvolvidos considerando suas características.

Hair et al [145] afirmam que o conhecimento sobre os dados que serão analisados é importante em um processo de análise. Entretanto, segundo Hebeler et al [149], este não é presente em muitos sistemas de informação. Esse conhecimento nem sempre é explícito e fácil de interpretar. Assim como na análise de dados tradicional, séries temporais podem ser semanticamente enriquecidas, onde informações de proveniência usando ontologias permitem representar e inferir conhecimento.

A Ontologia de Séries Temporais (prefixo tso:) é uma ontologia de domínio (um módulo em linguagem OWL) com a definição dos principais conceitos e relacionamentos envolvidos em proveniência de séries temporais. Essa ontologia adiciona conhecimento semântico aos dados, contribuindo para a escolha de métodos estatísticos apropriados para um importante passo da análise, que é a extração de tendências.

A Ontologia TSO descreve séries temporais relacionadas a processos não-estacionários, ou seja, apresentam tendências que necessitam ser extraídas [70], as quais são a regra e não a exceção em muitos domínios de aplicação. Em relação à extensibilidade, por um lado, a ontologia é extensível devido ao reuso de declarações a partir do conjunto SWEET, onde as declarações podem ser estendidas, quando aplicável, com base nessas ontologias e, por outro lado, permite extensão pela forma como a mesma foi projetada. A especificação dos requisitos da Ontologia para Proveniência de Séries Temporais é descrita a seguir, conforme os Cenários 1, 3 e 8 da Metodologia *NeOn* [233].

6.2.1 Documento de Especificação de Requisitos (TSO ORSD)

O ORSD contempla o propósito, o escopo, a linguagem de implementação, os usuários e os requisitos da ontologia, conforme segue.

1. Propósito: O propósito da ontologia TSO é descrever e inferir conhecimento semântico quanto à proveniência das séries temporais no passo de correção de tendências da fase de pré-processamento. Esse modelo é desenvolvido com base na literatura de Análise de Séries Temporais.

2. Escopo: A ontologia apresenta a definição dos principais conceitos e relacionamentos inerentes às séries temporais, incluindo suposições, modelos e tipos de decomposição, componentes e suas características, entre outras informações. Em relação ao escopo, não são incluídos métodos estatísticos para correção das séries temporais, os quais são definidos em outra ontologia.

3. Linguagem de Implementação: A ontologia é implementada na linguagem OWL, sendo utilizadas as linguagens SPARQL para desenvolvimento de consultas e SWRL para desenvolvimento de regras.

4. Usuários Finais Pretendidos: Os usuários finais são os pesquisadores de análise de séries temporais e proveniência de dados.

5. Usos Pretendidos: Envolvem o desenvolvimento de consultas ricas semanticamente quanto à geração de informações de proveniência relacionadas às séries temporais, envolvidas no passo de extração de tendências.

6. Requisitos da Ontologia: Dividem-se em:

   - Requisitos Não-Funcionais (RNF):
     RNF1. A terminologia usada na ontologia TSO é definida com base na literatura de Análise de Séries Temporais.

     RNF2. A ontologia é definida no idioma Inglês.

     RNF3. A ontologia é escrita seguindo a convenção de nomenclatura: nomes de classes são criados em maiúscula, sem usar caracteres alfa-numéricos ou especiais, usando singular sempre que possível; relacionamentos iniciam com minúscula e as demais palavras com a inicial maiúscula. Sempre que possível, nos relacionamentos, é utilizado a convenção (*hasProperty*), por exemplo, (*hasKnowledgeDomain*); instâncias iniciam com maiúsculas e usam *underline* para envolver palavras conjuntas, por exemplo (*Time_Series_Analysis*).
   - Requisitos Funcionais (RF): Para definição da ontologia, foi identificado, juntamente com especialistas, um conjunto de questões de competência que a ontologia deveria ser capaz de responder, envolvendo características intrínsecas sobre dados de séries temporais e seus componentes. Essas questões foram identificadas baseadas no modelo conceitual W7 [215], as quais encontram-se descritas na sequência. A partir das questões de competência, foi extraído um pré-glossário de termos, o qual é estendido na definição da ontologia.

Na sequência são descritas as questões de competência levantadas para o desenvolvimento da ontologia para proveniência de séries temporais não-estacionárias.

Questões de Competência (QCs) - Ontologia TSO

QC1. Questões *What*? - um evento que afetou as séries temporais

Evento: Geração das séries temporais

1. Qual o tipo de observação das séries temporais?
2. Qual o intervalo em que as séries temporais regularmente espaçadas foram observadas (diário/semanal/de hora em hora, entre outros)?
3. Qual a porcentagem de dados ausentes (*missing data*) das séries temporais?
4. Quais componentes de evento extremo as séries temporais observadas apresentam (*outliers*, *jumps*, entre outros)?
5. Qual propriedade matemática as séries temporais apresentam? (tempo discreto/contínuo)

6. Quais são as suposições consideradas pelo pesquisador quanto às séries temporais (homogeneidade, linearidade, entre outras)?
7. Qual é a área de conhecimento das séries temporais?
8. Qual a coleção que as séries temporais são originárias?
9. Qual a função de Auto-correlação das séries temporais?
10. Qual é o sinal de Auto-correlação das séries temporais observadas (negativamente/positivamente autocorrelacionadas)?
11. Qual a análise feita nas séries temporais (Univariada, Bivariada)?

Evento: Decomposição das séries temporais

1. Qual o modelo de decomposição das séries temporais (aditivo/multiplicativo, entre outros)?
2. Qual tipo de decomposição é feita nas séries temporais (clássica, entre outros)?
3. Quais componentes o modelo de decomposição apresenta (tendência mais irregular, entre outros)?
4. Qual análise é relacionada aos componentes das séries temporais?

QC2. Questões *When*? - refere-se ao tempo em que o evento ocorreu

Evento: Geração das séries temporais

1. Quando é inicializada/finalizada a observação das séries temporais?

QC3. Questões *Where*? - localização do evento

Evento: Localização do armazenamento das séries temporais

1. Onde as séries temporais são armazenadas?

QC4. Questões *How*? - ação que conduziu ao evento

Ação: Observação das séries temporais
Evento: Classificação das séries temporais

1. Como podem ser classificadas as séries temporais conforme as suposições (linearidade, normalidade, homoscedasticidade, entre outras), conforme o tipo de propriedade matemática relacionada (por exemplo, tempo discreto) ou conforme o tipo de observação feita?
2. Como podem ser classificadas as séries temporais conforme o tipo de dado relacionado (dados brutos, transformados, entre outros)?
3. Como podem ser classificadas as séries temporais conforme a não-estacionariedade (média, variância ou em ambas)?

Ação: Observação das séries temporais
Evento: Geração das séries temporais

1. Como as séries temporais foram geradas (processo estocástico estacionário /não-estacionário)?

Ação: Inserção de séries temporais
Evento: Armazenamento das séries temporais

1. Como as séries temporais são identificadas?

QC5. Questões *Which*? - programas ou instrumentos utilizados no evento

Agente: Software, Instrumento científico
Evento: Geração das séries temporais

1. Qual programa ou instrumento científico gerou as séries temporais?

As questões a seguir são relacionadas à proveniência do componente tendência das séries temporais não-estacionárias.

QC1. Questões *What*? - um evento que afetou as séries temporais

Evento: Geração das séries temporais

1. Qual o período de tempo relacionado à tendência (longo/curto prazo)?

Evento: Modelagem das séries temporais

1. Qual tendência é considerada na modelagem das séries temporais (determinística, estocástica, suavizada)?
2. Qual modelo de tendência determinística é considerado na modelagem (polinomial, *ring-width*, entre outros)?
3. Qual a forma de ajuste do modelo de tendência determinística (global ou local)?
4. Qual comportamento a tendência apresenta (monotônica/periódica/senoidal, entre outras)?

6.2.2 Reuso de Recursos Ontológicos

As atividades desenvolvidas para reuso de recursos ontológicos na Ontologia TSO são:

1. Pesquisa de ontologias. Para realizar esta atividade, pesquisas por recursos ontológicos que satisfaçam os requisitos da ontologia foram efetuadas em repositórios e registros tais como Swoogle [53] e Watson [57], assim como utilizando o *Plugin* Watson [56], a partir da ferramenta *NeOn* [233].

   Na ontologia de séries temporais, o reuso foi considerado como forma de interoperabilidade semântica. Para seu desenvolvimento, as seguintes ontologias foram analisadas quanto à possibilidade de reuso:

- Ontologia de Observações de Séries Temporais, uma representação ontológica de observações de séries temporais baseada no modelo O&M XML [150], especificando as várias formas de observações das séries temporais.
- Ontologias SWEETAll.owl.

2. Avaliação da ontologia. Nessa atividade, os recursos ontológicos obtidos na atividade anterior são inspecionados quanto ao conteúdo e granularidade, objetivando verificar se tais recursos satisfazem os requisitos de ORSD.

3. Comparação de ontologias. Os recursos ontológicos são comparados conforme custos econômicos, clareza do código e qualidade do conteúdo.

   Em relação às atividades acima, pelos seguintes motivos as ontologias analisadas não foram consideradas para reutilização. Essa conclusão envolveu inclusive a participação de especialistas do domínio:

   - A URL da ontologia não está disponível.
   - A ontologia é demasiadamente extensa para reuso.
   - A ontologia modela os conceitos de forma mais genérica do que o contexto necessário.

4. Seleção de ontologia. Desenvolvedores devem selecionar o conjunto de recursos ontológicos que são apropriados, baseados nos critérios definidos.

   Devido ao fato de que até o momento não foi encontrada uma ontologia descrevendo séries temporais e suas características, as quais são consideradas relevantes para tomada de decisão pelo pesquisador quanto a um processo de extração de tendências, no desenvolvimento da ontologia TSO, optou-se pelo reuso de declarações a partir do conjunto de Ontologias SWEETAll.owl, Versão 2.3. Uma descrição mais detalhada sobre as ontologias SWEET encontra-se na Seção 6.3.2 deste documento.

   Após a definição dos recursos ontológicos, o reuso considera os mesmos conforme se encontram na ontologia original.

5. Integração de ontologia. Desenvolvedores de ontologia devem incluir os recursos ontológicos selecionados, por meio das atividades do Cenário 1.

   Como resultado deste Cenário, têm-se uma ontologia implementada, assim como um conjunto de documentos relacionados às diferentes atividades. A próxima seção aborda a definição da ontologia TSO.

6.2.3 Definição da Ontologia TSO

Após especificar os Cenários 1 e 3, o Cenário 8 é relacionado com as atividades de modularização, corte e enriquecimento da ontologia, envolvendo a extensão e/ou especialização de conceitos. A ontologia TSO foi projetada para ser desenvolvida de forma modular, não sendo necessário a modularização de recursos ontológicos. Em sua definição, é considerado o reuso de declarações, a partir do conjunto de ontologias SWEET, dada a dimensão para se importar essa ontologia como um todo. A partir do reuso de declarações semânticas, a ontologia TSO foi estendida para modelagem das características intrínsecas das séries temporais, de seus componentes, formas de decomposição, entre outras informações. A partir do reuso, determinadas declarações foram especializadas, assim como novas declarações foram especificadas, baseadas em fundamentação teórica da análise de séries temporais. A saída do Cenário 8 é a ontologia que representa o domínio esperado, a qual é implementada na linguagem OWL. Na sequência é apresentada a definição das classes, relacionamentos e objetos da ontologia TSO.

A partir das questões de competência, as classes e relacionamentos foram identificados, assim como algumas instâncias. Restrições nas classes e relacionamentos são declaradas usando axiomas e/ou regras, adicionando semântica e permitindo inferências.

O diagrama da Figura 6.1 apresenta as classes relacionadas com a classe (tso:TimeSeriesData) e os respectivos elementos de proveniência do Modelo W7.

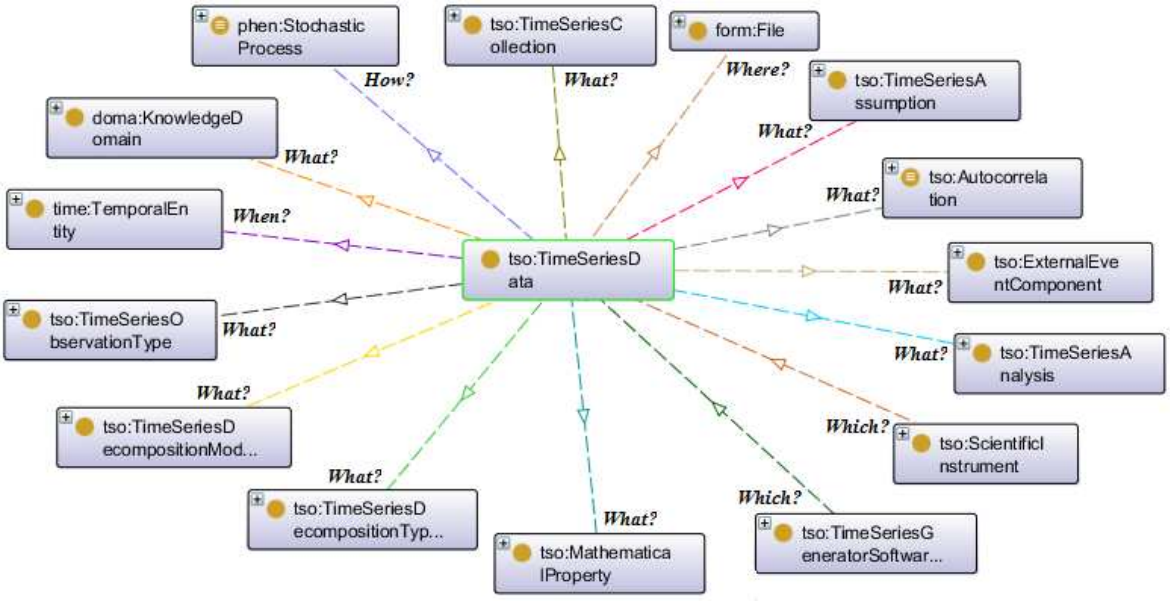


Figura 6.1: Classes TSO e associação com o modelo conceitual W7.

As classes (Figura 6.1) representam as definições e características das séries temporais, incluindo: processos que geraram as séries e tempo relacionado; tipo de análise associada e suposições consideradas pelo pesquisador em uma análise exploratória dos dados; tipo de observação; como as séries temporais podem ser classificadas de acordo com as suposições; domínio do conhecimento; coleção; instrumento científico ou software gerador; propriedade matemática associada; função de auto-correlação; arquivo relacionado; modelos e tipos de decomposição das séries temporais e seus componentes, assim como os componentes de evento extremo. A Figura 6.2 descreve os axiomas definidos na classe (tso:TimeSeriesData).

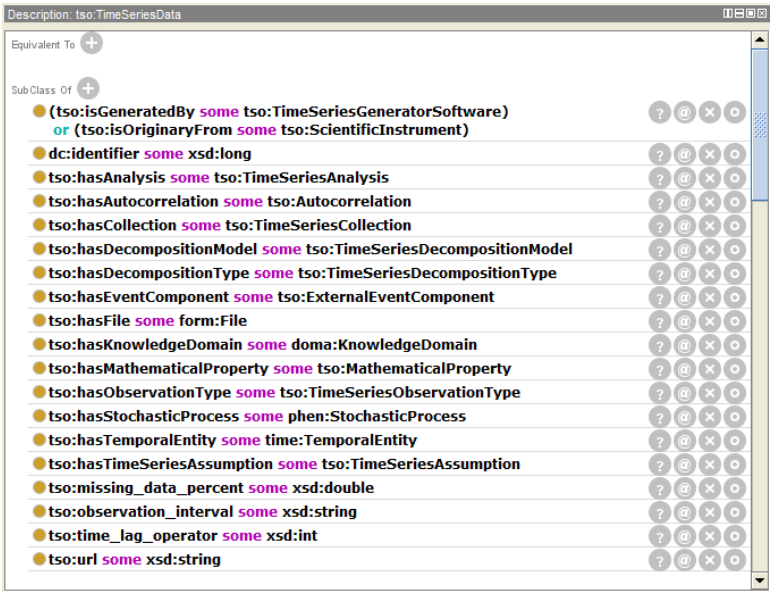


Figura 6.2: Classe (tso:TimeSeriesData) e axiomas definidos.

A Figura 6.3 apresenta o Diagrama de Classes da ontologia, contendo a descrição quanto à geração, modelagem e armazenamento das séries temporais, a partir de (tso:TimeSeriesData). A Figura 6.4 apresenta a descrição dos componentes, incluindo componentes de evento extremo, o modelo e tipo de decomposição das séries temporais. As propriedades de dados incluídas são: (*dc:identifier*), um identificador único para as séries; (*missing_data_percent*), representando a porcentagem de dados ausentes nas séries temporais; *(url)*, indicando sua localização; e (*time_lag_operator*), indicando o valor do operador *lag*.

A Figura 6.5 apresenta a especialização da classe (tso:TimeSeries) nas respectivas subclasses, e a Figura 6.6 apresenta as respectivas classes declaradas como disjuntas. Nesta classe são feitas inferências pelo *reasoner* Pellet, em relação às suposições consideradas pelo pesquisador, tipo de observação feita e a propriedade matemática associada.

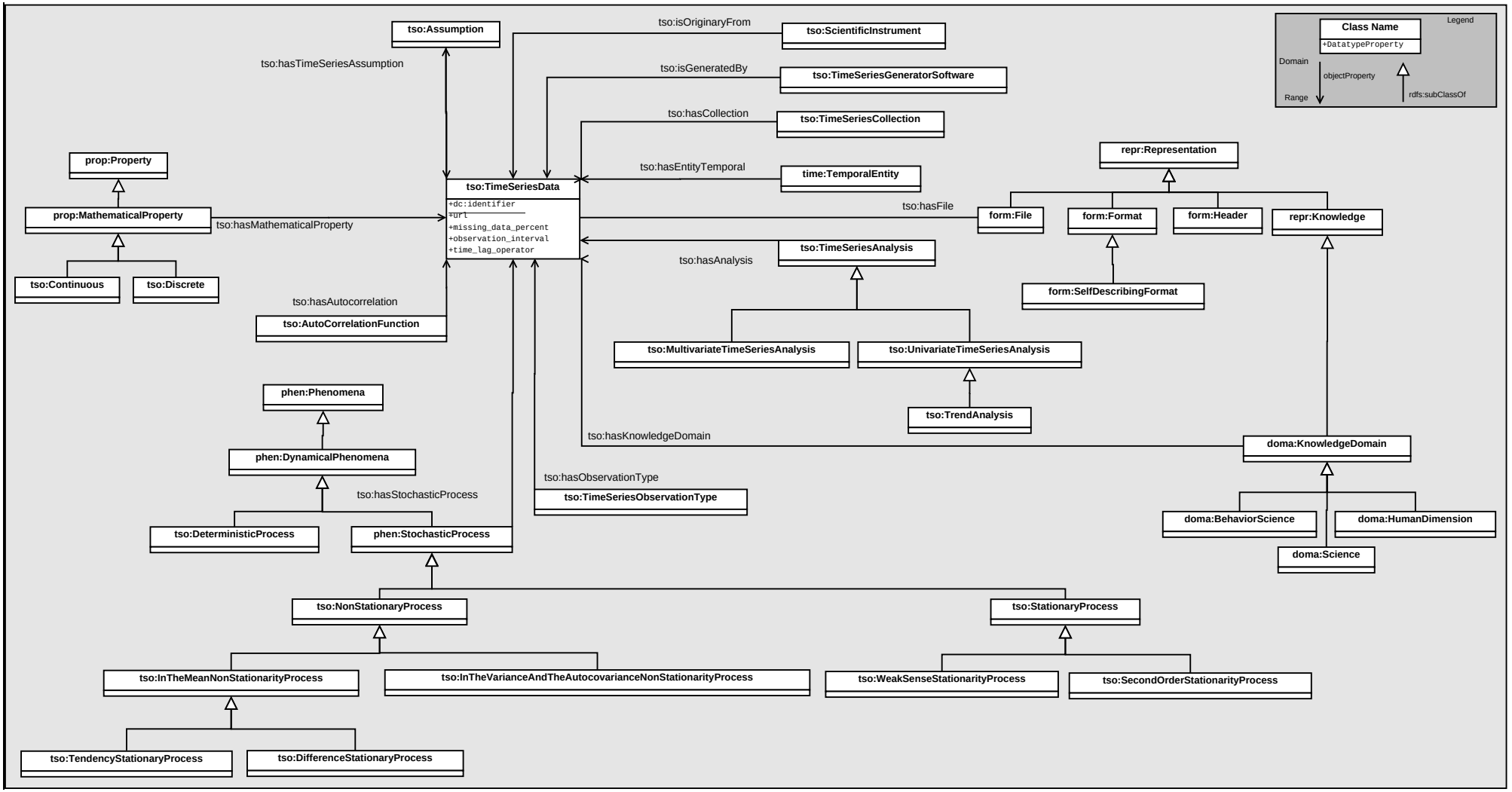


Figura 6.3: Classe (tso:TimeSeriesData) e relacionamentos.

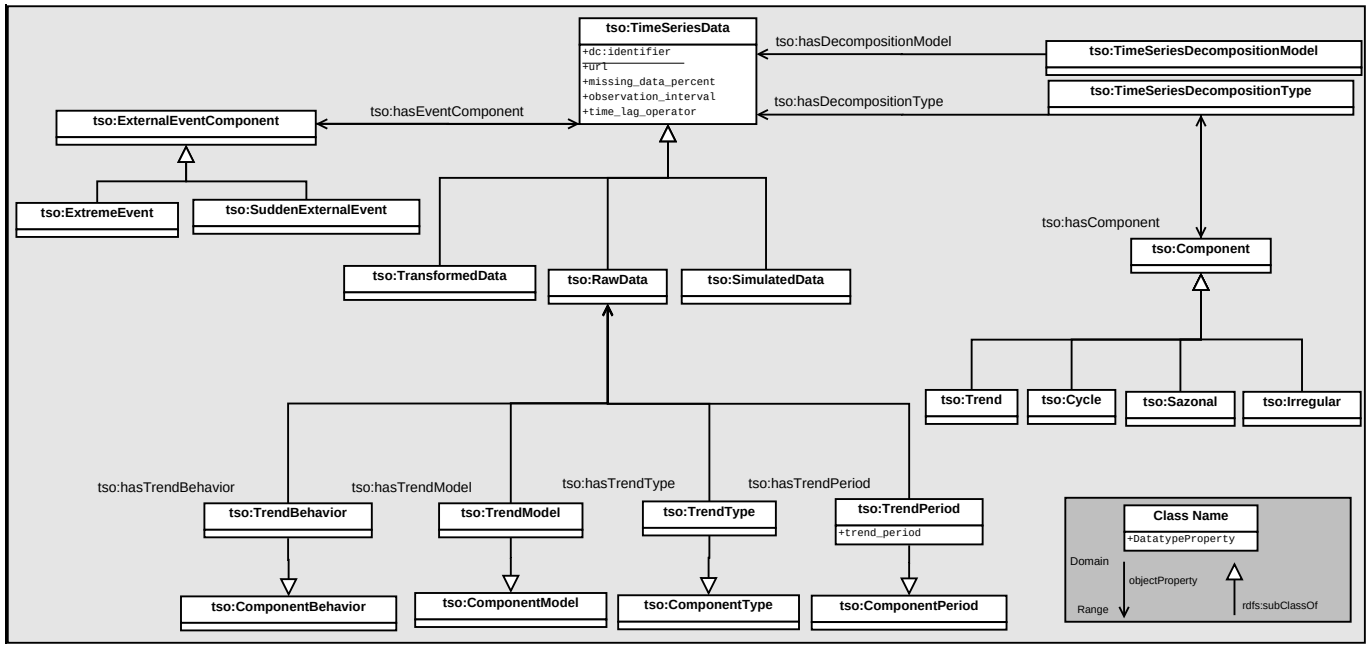


Figura 6.4: Classe (tso:TimeSeriesData) e decomposição das séries temporais.

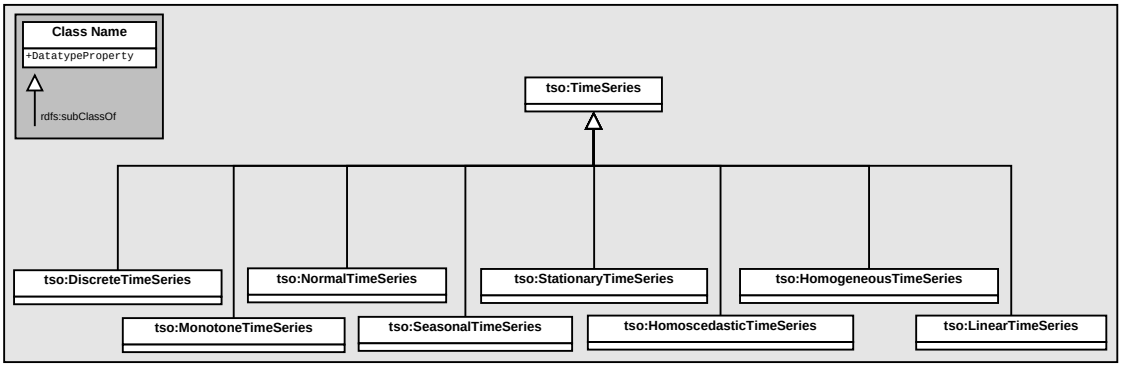


Figura 6.5: Classe (tso:TimeSeries) e subclasses.

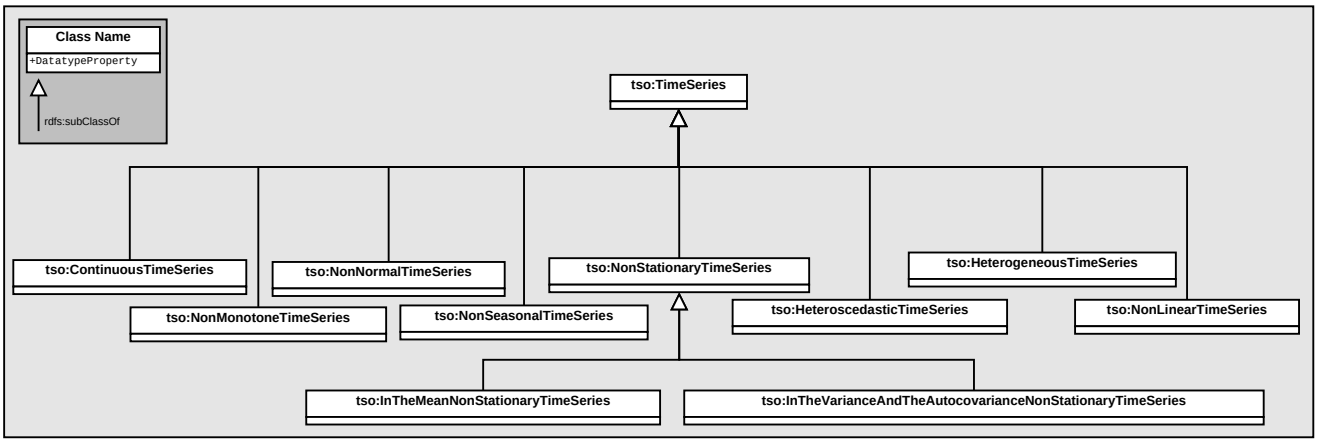


Figura 6.6: Classe (tso:TimeSeries) e subclasses disjuntas.

A ontologia faz uma classificação das séries temporais usando regras definidas na Linguagem SWRL. Abaixo são apresentados alguns exemplos de regras definidas, as quais permitem inferir conhecimento semântico sobre as séries temporais, contribuindo com o processo de análise dos dados. Inferências são feitas pelo *reasoner* Pellet [227].

(1) tso:TimeSeriesData(?x), tso:hasObservationType(?x, tso:Regularly_Spaced) → tso:HomogeneousTimeSeries(?x)

(2) tso:TimeSeriesData(?x), tso:hasObservationType(?x, tso:Irregularly_Spaced) → tso:HeterogeneousTimeSeries(?x)

(3) tso:TimeSeriesData(?x), tso:hasTimeSeriesAssumption(?x, tso:Linearity) → tso:LinearTimeSeries(?x)

(4) tso:TimeSeriesData(?x), tso:hasTimeSeriesAssumption(?x, tso:NonNormality) → tso:NonNormalTimeSeries(?x)

(5) tso:TimeSeriesData(?x), tso:hasTrendType(?x, tso:Deterministic_Trend) → tso:NonStationaryTimeSeries(?x)

As duas primeiras regras são relacionadas com o tipo de observação (regularmente ou irregularmente espaçadas) e sua classificação, de acordo com o tipo das séries relacionadas: homogêneas ou heterogêneas, conforme [183, 82]. A terceira regra é relacionada à suposição de linearidade, a qual permite inferir tais séries temporais como lineares. O mesmo ocorre na quarta regra, com a suposição de não-normalidade, onde as mesmas são inferidas como séries temporais não-normais. Assim como estas, outras regras classificam as séries temporais nas devidas classes, conforme suposições declaradas pelo pesquisador. Outro

caso do desenvolvimento de regras é quanto à propriedade matemática associada, por exemplo, se é declarado que as séries são de tempo discreto, as mesmas são inferidas como sendo da classe (tso:DiscreteTimeSeries).

Quando as séries temporais apresentam algum tipo de tendência, por exemplo, determinística, conforme a quinta regra, as mesmas são inferidas na classe (tso:NonStationaryTimeSeries), onde a tendência necessita ser extraída porque pode ocultar outros fenômenos, assim como a maioria dos métodos estatísticos são desenvolvidos considerando estacionariedade.

Outro caso de regras é quando é declarado que a série temporal é gerada por um processo não-estacionário na média, as séries são inferidas na respectiva classe de séries temporais não-estacionárias na média.

Na ontologia TSO, a partir do conjunto de Ontologias SWEET, são reutilizados as declarações (phen:Phenomena) e suas subclasses (phen:DynamicalPhenomena) e (phen:StochasticProcess) e sua propriedade de objeto (rela:hasPhenomena). Em TSO, a classe (phen:StochasticProcess) é declarada como uma classe disjunta de (tso:NonStochasticProcess). A propriedade de objeto reutilizada é estendida em (tso:hasDynamicalPhenomena) e especializada nas propriedades disjuntas (tso:hasStochasticProcess) e (tso:hasNonStochasticProcess).

Conforme uma regra definida, se a não-estacionariedade ocorre na média e o processo gerador das séries é considerado de diferença estacionária, a inferência feita sobre a tendência é (tso:Stochastic_Trend). Da mesma forma, se o processo gerador das séries temporais é de tendência estacionária, a inferência feita sobre o tipo de tendência relacionada é (tso:Deterministic_Trend) [232].

A informação sobre o tipo da tendência é relevante para a escolha dos métodos estatísticos apropriados para correção das séries temporais. No caso da tendência ser considerada como estocástica, a mesma pode ser removida pelo método *Differencing*. No caso da tendência ser considerada como determinística, a mesma pode ser estimada por um ajuste de um método determinístico, como análise de regressão paramétrica, ajustando um modelo para a tendência que é posteriormente removida. Na sequência, são apresentados casos de uso relacionados à proveniência de séries temporais, contribuindo quanto à tomada de decisão sobre o uso de um método de *detrending* apropriado.

6.2.4 Descrição de Casos de Uso e Consultas Relacionadas

Os casos de uso identificados e que motivam o desenvolvimento da Ontologia TSO são descritos na sequência.

Caso de Uso 1: Em um processo de análise, é importante para o pesquisador saber se as séries temporais discretas não-estacionárias, ou seja, possuem tendências, apresentam eventos extremos, tais como *outliers* [145]. Essa informação contribui para a tomada de decisão sobre qual método de *detrending* utilizar.

A Figura 6.7 apresenta uma consulta relacionada ao Caso de Uso 1, mostrando as séries temporais e suas características, incluindo a informação se as mesmas apresentam algum componente de evento extremo. Essa informação contribui para o pesquisador escolher métodos estatísticos apropriados para uso, como um método robusto, que é resistente a *outliers* [95]. A associação das instâncias com a DBpedia permite gerar conhecimento, além de contribuir para interoperabilidade semântica (Figura 6.8). Por meio desta associação, a partir da DBpedia, é possível obter informações usando outras

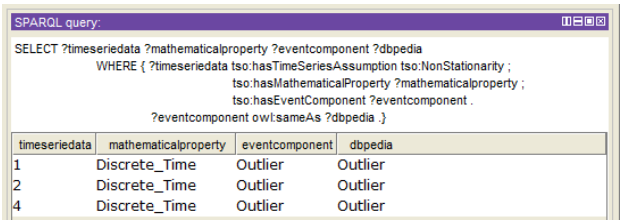


Figura 6.7: Caso de Uso 1. Características de séries temporais não-estacionárias mostrando eventos extremos, associados com a DBpedia.

tags, tais como (dbpedia-owl:abstract), apresentando um resumo; (foaf:primaryTopic), a qual associa com a respectiva página na Wikipedia; e (foaf:depiction), associando o termo a uma imagem, entre outras informações disponíveis pelas *tags*.

Figura 6.8: Associação de instância com dbpedia:*Outlier*.

Outra situação deste caso de uso é em relação ao conhecimento, se as séries apresentam eventos extremos tais como *jumps* (saltos aleatórios), contribuindo para a tomada de decisão quanto a utilizar um método que considere ou não a presença de tais componentes.

Caso de Uso 2: esse caso de uso é relacionado à identificação do método estatístico apropriado considerando o tipo de observação (regularmente ou irregularmente espaçada). Em um processo de análise, a função de Auto-Correlação necessita de múltiplos pares de observações para quantificar a dependência serial. Nesse caso, as séries temporais necessitam ser regularmente espaçadas e com poucos dados ausentes. Essa abordagem não é apropriada para séries com intervalos altamente irregulares, necessitando de técnicas alternativas [95]. Na sequência são apresentados mais alguns cenários onde o conhecimento sobre o tipo de observação é relevante em um processo de análise.

Rehfeld et al [218] apresentam uma comparação de técnicas de análise de correlação para séries temporais irregularmente amostradas, tais como medidas geo-científicas que apresentam, com frequência, séries temporais irregularmente espaçadas. Nesse caso, faz-se necessário utilizar métodos de reconstrução dos dados (interpolação) ou, usar métodos mais sofisticados, para tratar amostras irregulares. A amostra irregular de séries temporais torna o uso de técnicas de estimação padrão complicado, pois estas precisam de observações regulares [218].

Erdogan et al [122] apresentam um *framework* teórico para analisar séries temporais irregularmente espaçadas, ambas estacionárias e não-estacionárias. É citado que a maioria dos métodos trata séries regularmente espaçadas, não sendo facilmente estendidos para dados irregularmente amostrados. Na prática da análise de séries temporais, a irregularidade é uma característica dos dados, onde pesquisadores tratam essa questão de forma heurística. Uma prática comum citada pelos autores é ignorar essa situação e tratar os dados como se fossem regulares, mas isso pode introduzir um significante *bias*, levando a predições incorretas, visto que tal operação muda a dinâmica do processo. Questões bem-entendidas para séries regulares não se aplicam a séries irregularmente amostradas, onde muitas técnicas tratam séries com dados ausentes (*missing data*), as quais em um limite, podem ser vistas como uma amostra irregularmente espaçada.

Eckner [119] descreve métodos de estimação dos componentes tendência, sazonal e irregular de séries temporais univariadas irregularmente espaçadas, citando que existem muitos métodos para amostras regularmente espaçadas e, por outro lado, poucos métodos existem para séries especificamente irregularmente espaçadas, apesar de estes dados ocorrem naturalmente, em muitos domínios.

Por outro lado, o método de média móvel [223], é somente aplicado para séries regularmente espaçadas. Dessa forma, o conhecimento sobre o tipo de observação das séries contribui na tomada de decisão sobre qual método utilizar em cada caso.

SPARQL query:

```
SELECT ?timeseriesdata ?missingdata ?observationinterval ?mathematicalproperty ?observationtype
        WHERE { ?timeseriesdata tso:missing_data_percent ?missingdata ;
                                tso:observation_interval ?observationinterval ;
                                tso:hasMathematicalProperty ?mathematicalproperty ;
                                tso:hasObservationType ?observationtype . }
```

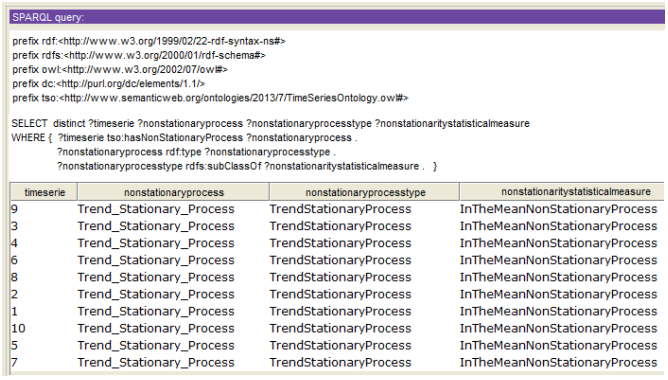
timeseriesdata	missingdata	observationinterval	mathematicalproperty	observationtype
1	"0.02"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced
4	"0.02"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced
8	"0.04"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced
9	"0.05"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced
3	"0.02"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced
7	"0.02"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced
6	"0.02"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced
2	"0.02"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced
10	"0.02"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced
5	"0.02"^^<	"512 sec"^^<	Discrete_Time	Regularly_Spaced

Figura 6.9: Caso de Uso 2. Porcentagem de dados ausentes, intervalo de observação, propriedade matemática e tipo de observação das séries temporais.

A Figura 6.9 mostra, além do tipo de observação das séries temporais, algumas características das mesmas, tais como porcentagem de dados ausentes, o intervalo de observação e a propriedade matemática associada. Essas informações contribuem para a tomada de decisão pelo pesquisador sobre qual método utilizar nas séries temporais, como por exemplo, quantificar a função de Auto-Correlação, a qual mede a dependência entre sucessivas observações.

Caso de Uso 3: é importante o conhecimento sobre o processo gerador das séries temporais e em qual medida estatística a não-estacionariedade ocorre, ou seja, na média ou na variância. A não-estacionariedade na média pode ser removida pelo método de *Differencing*, entre outros métodos e a não-estacionariedade na variância necessita de outras transformações nos dados [244].

No Caso de Uso da Figura 6.10, o conhecimento sobre qual processo e medida estatística a não-estacionariedade ocorre, permite escolher um método apropriado para sua extração considerando a não-estacionariedade na média das séries temporais.



SPARQL query:

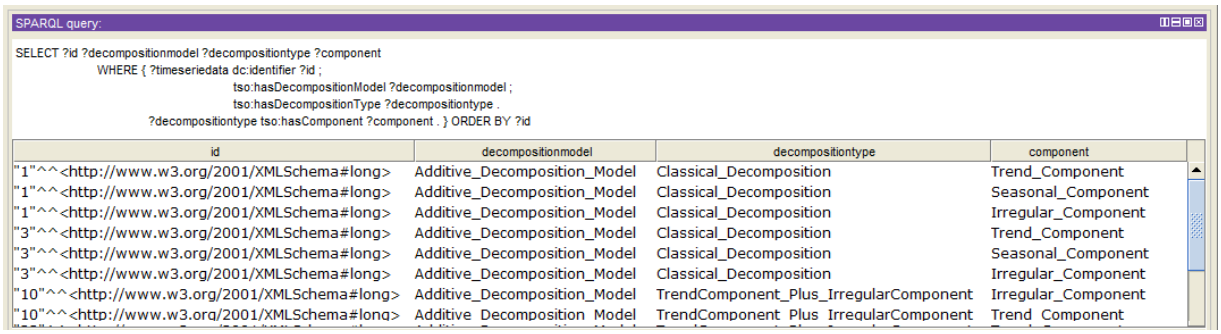
```
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix owl:<http://www.w3.org/2002/07/owl#>
prefix dc:<http://purl.org/dc/elements/1.1/>
prefix tso:<http://www.semanticweb.org/ontologies/2013/7/TimeSeriesOntology.owl#>

SELECT  distinct ?timeserie ?nonstationaryprocess ?nonstationaryprocesstype ?nonstationaritystatisticalmeasure
WHERE {  ?timeserie tso:hasNonStationaryProcess ?nonstationaryprocess .
         ?nonstationaryprocess rdf:type ?nonstationaryprocesstype .
         ?nonstationaryprocesstype rdfs:subClassOf ?nonstationaritystatisticalmeasure .  }
```

timeserie	nonstationaryprocess	nonstationaryprocesstype	nonstationaritystatisticalmeasure
9	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess
3	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess
4	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess
6	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess
8	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess
2	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess
1	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess
10	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess
5	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess
7	Trend_Stationary_Process	TrendStationaryProcess	InTheMeanNonStationaryProcess

Figura 6.10: Caso de Uso 3. Processo gerador e em qual medida estatística ocorre a não-estacionariedade.

Caso de Uso 4: esse caso de uso é relacionado com o conhecimento sobre o modelo de decomposição das séries. Essa informação contribui para o pesquisador escolher métodos estatísticos apropriados para remoção da tendência das séries temporais. Para exemplificar, se as séries temporais são consideradas em um modelo aditivo, a tendência estimada é subtraída a partir dos dados originais [249]. Em um modelo de decomposição multiplicativo, a mesma é removida pela divisão das séries temporais originais pelos valores da tendência estimada.



SPARQL query:

```
SELECT ?id ?decompositionmodel ?decompositiontype ?component
        WHERE { ?timeseriedata dc:identifier ?id ;
                               tso:hasDecompositionModel ?decompositionmodel ;
                               tso:hasDecompositionType ?decompositiontype .
                ?decompositiontype tso:hasComponent ?component . } ORDER BY ?id
```

id	decompositionmodel	decompositiontype	component
"1"^^<http://www.w3.org/2001/XMLSchema#long>	Additive_Decomposition_Model	Classical_Decomposition	Trend_Component
"1"^^<http://www.w3.org/2001/XMLSchema#long>	Additive_Decomposition_Model	Classical_Decomposition	Seasonal_Component
"1"^^<http://www.w3.org/2001/XMLSchema#long>	Additive_Decomposition_Model	Classical_Decomposition	Irregular_Component
"3"^^<http://www.w3.org/2001/XMLSchema#long>	Additive_Decomposition_Model	Classical_Decomposition	Trend_Component
"3"^^<http://www.w3.org/2001/XMLSchema#long>	Additive_Decomposition_Model	Classical_Decomposition	Seasonal_Component
"3"^^<http://www.w3.org/2001/XMLSchema#long>	Additive_Decomposition_Model	Classical_Decomposition	Irregular_Component
"10"^^<http://www.w3.org/2001/XMLSchema#long>	Additive_Decomposition_Model	TrendComponent_Plus_IrregularComponent	Irregular_Component
"10"^^<http://www.w3.org/2001/XMLSchema#long>	Additive_Decomposition_Model	TrendComponent_Plus_IrregularComponent	Trend_Component

Figura 6.11: Casos de Uso 4 e 5. Modelo e tipo de decomposição e componentes das séries temporais.

A Figura 6.11 apresenta o modelo e o tipo de decomposição considerado nas séries temporais. O conhecimento sobre a decomposição contribui quanto à forma de como a tendência pode ser removida das séries. Nesse caso, por subtração, devido ao modelo de decomposição aditivo.

Caso de Uso 5: o conhecimento se as séries apresentam ruído de alta frequência é importante quanto a tomada de decisão quanto a utilizar um método de filtragem do mesmo, gerando dados filtrados, antes do passo de *detrending*. A Figura 6.11 apresenta os componentes das séries temporais, as quais apresentam o componente Irregular (Ruído), contribuindo para a escolha de métodos que considerem esse componente. O mesmo ocorre em relação ao conhecimento sobre se a série temporal apresenta um componente sazonal, sendo relevante para o pesquisador considerar se as mesmas necessitarão de correção de sazonalidade.

Caso de Uso 6: o conhecimento sobre o modelo de tendência considerado é relevante para a tomada de decisão quanto a utilizar, por exemplo, um filtro em que tendências passarão, por exemplo, se a tendência é considerada linear é adequado usar um filtro linear para identificação da mesma, conforme [95] (Figura 6.12).

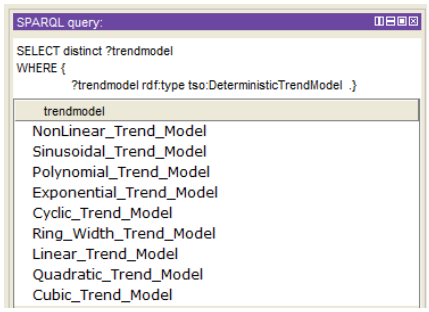


Figura 6.12: Caso de Uso 6. Modelos de tendência.

Da mesma forma, o conhecimento sobre o tipo e o modelo de tendência é relevante para a tomada de decisão quanto ao método estatístico a utilizar para *detrending*. Por exemplo, se o tipo da tendência é considerado como determinístico e a forma de ajuste é global, nesse caso é possível usar um método de regressão, ajustando uma função aos dados que será removida a partir destes. Por outro lado, se o tipo da tendência é considerado intrínseco, onde o modelo necessita ser adaptativo aos dados, é possível fazer uso de um método de decomposição empírica das séries, como o método *Empirical Mode Decomposition* - EMD [156, 193].

Caso de Uso 7: o conhecimento sobre o tipo das séries temporais contribui para a escolha de um método adequado para *detrending*. Por exemplo, em séries temporais declaradas como não-estacionárias e não-lineares pode ser aplicado um método que decompõe as séries em funções de modo intrínsecas aos dados, como o método EMD [156] ou, um método de decomposição em componentes oscilatórios, como *Singular Spectrum Analysis* - SSA [192, 121].

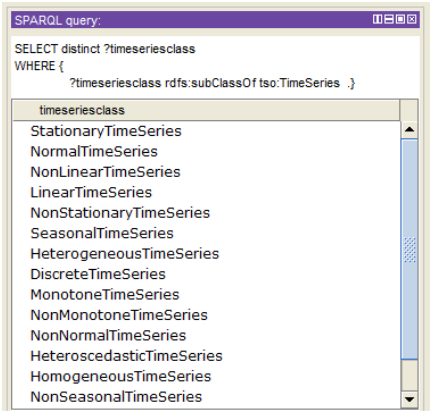


Figura 6.13: Caso de Uso 7. Classes de séries temporais.

Da mesma forma, se os dados são declarados como não normais, ou seja, não seguem uma distribuição Normal, essa informação contribui para a escolha de qual modelo ajustar

nos dados. Nesse caso, pode ser usado um Modelo Linear Generalizado - GLM, onde a suposição de normalidade é relaxada [95]. A Figura 6.13 mostra as subclasses de (tso:TimeSeries).

A especificação dos requisitos da Ontologia *Detrend* é descrita na sequência, incluindo os Cenários 1, 3 e 8 relacionados à Metodologia *NeOn* [233].

6.3 *Detrend Ontology* (DO)

6.3.1 Documento de Especificação de Requisitos (DO ORSD)

O ORSD contempla o propósito, o escopo, a linguagem de implementação, os usuários e os requisitos da ontologia, conforme segue.

1. Propósito: O propósito de desenvolver a Ontologia *Detrend* é descrever e inferir conhecimento semântico, quanto aos métodos de correção de tendências, os quais podem ser usados na fase de pré-processamento de séries temporais. Esse modelo é desenvolvido com base na literatura de Análise de Séries Temporais.

2. Escopo: Os métodos descritos na Ontologia *Detrend* relacionam-se com o domínio do tempo, podendo ser expandidos para o domínio da frequência.

3. Linguagem de Implementação: A ontologia é implementada na linguagem OWL, sendo utilizada a linguagem SPARQL para o desenvolvimento de consultas.

4. Usuários Finais Pretendidos: Os usuários finais são os pesquisadores de análise de séries temporais e proveniência de dados.

5. Usos Pretendidos: Envolvem o desenvolvimento de consultas ricas semanticamente quanto à geração de informações de proveniência no passo de extração de tendências.

6. Requisitos da Ontologia: Dividem-se em:

   - Requisitos Não-Funcionais (RNF):
     RNF1. A terminologia usada na Ontologia *Detrend* é baseada em referências bibliográficas da área de séries temporais.

     RNF2. A ontologia considera o idioma Inglês.

     RNF3. A ontologia é escrita seguindo a convenção de nomenclatura: nomes de classes são criados em maiúscula, no estilo de classes Java, sem usar caracteres alfa-numéricos ou especiais e usar espaços, usando singular sempre que possível; relacionamentos iniciam com minúscula e as demais palavras com a inicial maiúscula. Sempre que possível, nos relacionamentos, utilizar a convenção (*hasProperty*) e sua propriedade inversa como *isPropertyOf*, por exemplo, (*hasDomain*); instâncias iniciam com maiúsculas e usam *underline* quando envolver palavras conjuntas, por exemplo (*Time_Domain*).

- Requisitos Funcionais (RF): envolvem as Questões de Competência da Ontologia *Detrend* e encontram-se descritos, segundo seis grupos principais, como segue. A partir das questões de competência, foi extraído um pré-glossário de termos para desenvolvimento da ontologia.

Questões de Competência (QCs) - Ontologia DO

QC1. Questões genéricas

(a) Em qual fase da análise das séries temporais ocorre o passo de extração de tendências (*detrending*)?

(b) Quais métodos e parâmetros são usados pelos respectivos algoritmos de *detrending*?

(c) Qual a aplicabilidade dos métodos nos respectivos algoritmos de *detrending*?

(d) Qual o domínio dos métodos usados para *detrending*?

(e) Quais os respectivos algoritmos relacionados aos softwares de *detrending*?

(f) Qual algoritmo/software de *detrending* é desenvolvido, a partir de outro algoritmo/software de *detrending*?

(g) Qual a Estatística relacionada aos métodos de *detrending* e sua aplicabilidade nos respectivos algoritmos de *detrending*?

(h) Como é removido o componente tendência das séries temporais pelos algoritmos de *detrending*?

QC2. Questões relacionadas à Análise de Regressão

(a) Que tipo de análise de regressão é feita (Univariada, Bivariada, Multivariada /Múltipla)?

(b) Quais variáveis estão envolvidas na regressão?

(c) Qual amostra está envolvida na regressão?

(d) Qual função é ajustada na regressão?

(e) Quais os graus de liberdade da regressão?

(f) Qual a qualidade de ajuste do modelo de regressão?

(g) Qual método de estimação de parâmetros é usado na regressão?

(h) Qual modelo é ajustado na regressão (linear, não-linear)?

(i) Quais as suposições do modelo de regressão?

(j) Na análise multivariada/múltipla, qual método de seleção de variáveis independentes é usado?

(k) Quais transformações são feitas nas variáveis dependente e independente na Análise Univariada?

(l) Qual teste de hipótese é feito na análise de regressão?

(m) O que é considerado como hipótese nula e alternativa, nos testes de hipótese?

(n) Qual o nível de significância dos testes de hipótese?

(o) Qual o sumário estatístico (média, entre outros) da amostra da regressão?

QC3. Questões específicas à Análise de Regressão Não-Paramétrica (baseada em métodos de suavização):

(a) Quais são os métodos e parâmetros usados para suavização baseada em regressão local?

(b) Quais são os métodos e parâmetros usados para suavização *kernel*?

(c) Quais são os métodos e parâmetros usados para suavização *nearest neighbor*?

(d) Quais são os métodos e parâmetros usados para suavização *spline*?

(e) Quais são os métodos e parâmetros usados para suavização baseada em filtro?

QC4. Questões específicas aos métodos de filtro:

(a) Qual o parâmetro de suavização (*bandwidth*) e qual o método usado para sua seleção?

(b) Como podem ser classificados os filtros (lineares/não-lineares)?

(c) Como os filtros são implementados (convolução/recursão)?

(d) Qual a respectiva banda de freqüência (passa baixa/passa alta frequência) dos filtros lineares?

QC5. Questões específicas aos métodos, algoritmos e softwares:

(a) Em qual paradigma/linguagem de programação são implementados os softwares de *detrending*?

(b) Como podem ser classificados os algoritmos/softwares de *detrending*, conforme os métodos relacionados?

(c) Quais os parâmetros dos algoritmos/softwares de *detrending*?

QC6. Questões específicas a métodos de filtro

(a) Quais informações podem ser geradas quanto ao uso de métodos de filtros, que podem ser usados para remoção do ruído das séries temporais, na fase de pré-processamento?

6.3.2 Reuso de Recursos Ontológicos

As atividades desenvolvidas para reuso de recursos ontológicos na Ontologia *Detrend* são:

1. Pesquisa de ontologias. Para realizar esta atividade, pesquisas por recursos ontológicos que satisfaçam os requisitos da ontologia foram efetuadas em repositórios e registros, tais como Swoogle [53] e Watson [57], assim como utilizando o *Plugin* Watson [56], a partir da ferramenta *NeOn* [233].

   No desenvolvimento da ontologia *Detrend*, o reuso foi considerado uma atividade relevante, como forma de interoperabilidade semântica. Em relação à modelagem existente de softwares para uma extensão/especialização para sofwares de *detrending*, as seguintes ontologias foram analisadas quanto à possibilidade de reuso:

- *The Software Ontology* (SWO *Ontology*) [41]
- *Core Software Ontology* (CSO *Ontology*) [7], contendo *Core Ontology of Software Components* (COSC) e *Core Ontology of Services* (COS)
- *Core Ontology of Programs and Software* (COPS *Ontology*) [168]
- *EvoOnt - A Software Evolution Ontology*, contendo (VOM, SOM e BOM.owl) [10]
- *SEON - Software Evolution ONtologies* (SEON) [35]

Quanto às demais classes, as ontologias abaixo foram analisadas:

- ACM *Ontologies* [4]
- *Semantic Web for Earth and Environmental Terminology* (SWEETAll.owl) [40]
- *Ontology Computer Science for Non-Computer Scientists. Project* LT4eL-(CSnCSv0.01Lex)[21]
- *reprMathStatistics Ontology* [29]
- *Sciflow: A Scientific Workflow Ontology* [31]

2. Avaliação da ontologia. Nessa atividade, os recursos ontológicos obtidos na atividade anterior são inspecionados, quanto ao conteúdo e granularidade, objetivando verificar se os mesmos satisfazem os requisitos de ORSD.

3. Comparação de ontologias. Os recursos ontológicos são comparados conforme custos econômicos, clareza do código e qualidade do conteúdo.

   Quanto às atividades acima, as ontologias foram analisadas considerando o reuso, com a participação de especialistas do domínio. As mesmas não foram consideradas para reuso na Ontologia *Detrend*, pelos motivos que seguem, os quais podem estar presentes, inclusive com mais de uma ocorrência, nas referidas ontologias.

   - Apesar da ontologia modelar softwares, é específica a um determinado domínio. Essa questão implica inclusive na utilização do prefixo da ontologia, que é específico. Nesse caso, cita-se que uma re-estruturação da ontologia seria necessária.
   - Existem problemas de inconsistência na ontologia, demandando tempo extra para análise e correção.
   - A URL da ontologia não está disponível e, mesmo após contato com desenvolvedores, não foi possível acesso à mesma para reuso.
   - A ontologia é voltada para um paradigma de programação específico, como Orientação a Objetos, apresentando conceitos e relacionamentos específicos, sendo possível a extensão para demais paradigmas, demandando tempo extra, tanto para entendimento da ontologia a ser reutilizada, quanto para sua extensão.
   - A ontologia foi desenvolvida em uma versão de ferramenta ontológica, não compatível com uma versão atualizada da mesma.
   - A ontologia foi desenvolvida com base em uma ontologia de fundamentação. Essa situação é considerada ideal, entretanto, existem dificuldades no entendimento dos termos fundamentados, demandando tempo para adequação na modelagem, inclusive para sua extensão.

- A ontologia encontra-se disponível em uma linguagem diferente ou conceitual, necessitando de tradução para OWL.
- A ontologia é demasiadamente extensa, onde o reuso está associado à utilização de termos considerados desnecessários ao domínio relacionado.
- Uso numérico para definição das classes.
- Ausência de anotações nas classes usando (*rdfs:comment*), o que dificulta o entendimento dos conceitos para adequação à ontologia sendo desenvolvida.

4. Seleção de ontologia. Desenvolvedores devem selecionar o conjunto de recursos ontológicos mais apropriados, baseados nos critérios definidos. Após análises, na sequência são descritas as ontologias e/ou declarações semânticas consideradas para reuso.

- ***Semantic Web for Earth and Environmental Terminology* (SWEETAll.owl)**

  Ontologias SWEET [40] estão publicamente disponíveis e a versão atual é a 2.3. Representam um conjunto bem-definido, de aproximadamente duzentas ontologias consideradas de nível intermediário (*middle level ontology*), desenvolvido de forma modular, constituindo uma terminologia da Web Semântica para Ciências da Terra e Ambientais.

  O projeto modular contempla oito ontologias principais, contendo conceitos de nível superior (Figura 6.14). É possível visualizar todas as ontologias acessando o arquivo (SWEETAll.owl), disponível em [40] para *download* ou, alternativamente, é possível reutilizar as ontologias individualmente. É possível adicionar uma ontologia específica de domínio, usando os componentes da SWEET para desenvolver a ontologia desejada.

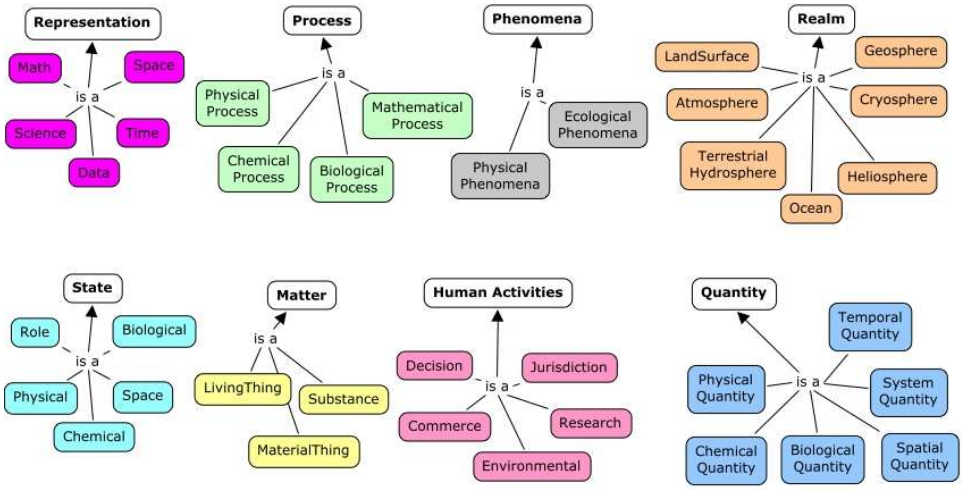


Figura 6.14: Conjunto de Ontologias SWEET [40].

No desenvolvimento de Ontologias SWEET, são citadas algumas diretrizes para o desenvolvimento de ontologias: escalabilidade, independência de aplicação e de linguagem natural, ortogonalidade de conceitos e envolvimento da comunidade [216].

Embora RDF e OWL não foram desenvolvidos para dar suporte a conceitos matemáticos porque os mesmos dependem da definição de esquema e são baseados na linguagem XML, o conjunto de ontologias de nível superior SWEET é um bom exemplo para integrar conhecimento matemático com domínios de aplicação científica, conforme [169].

DiGiuseppe et al [116] avaliam a cobertura do conjunto de ontologias SWEET [216] para o domínio de Ciências de Sistemas Ambientais e da Terra. Essa ontologia tem se tornado um padrão para representar formalmente esse domínio. É apresentado um estudo de aplicabilidade de técnicas de cobertura para uso dentro do domínio, assim como experimentos para avaliar empiricamente a relevância do uso de Ontologias SWEET. A metodologia proposta de avaliação considera, não somente a cobertura de termos, mas também os axiomas de subclasses, usando a abordagem *Synonym Synergy*. Essa abordagem usa um tesauro para aproximar relacionamentos de subclasses, usando relacionamentos sinônimos, onde a aproximação dos relacionamentos é feita pela determinação de quais combinações de palavras são somente sinônimos parciais, ou seja, um relacionamento sinônimo que não é bi-direcional é quando um termo A é sinônimo de B, mas o termo B não é sinônimo de A. Os resultados apontam que a metodologia proposta é aplicável e que as ontologias SWEET precisamente representam o referido domínio. Nessa tese, para o desenvolvimento das ontologias, algumas declarações OWL a partir de SWEETAll (Versão 2.3) consideradas pertinentes ao domínio foram reutilizadas e/ou estendidas.

Quanto ao reuso, após analisar a ontologia (reprMathStatistics) do conjunto SWEET (Figura 6.15), e considerando que a mesma importa indiretamente muitas outras ontologias e que não estariam relacionadas ao contexto, esta não foi considerada para modelagem da regressão, visto que necessita de extensões. A ontologia (StatisticalAnalysis) [36], descrita na sequência, estende as classes e relacionamentos, sendo reutilizada e estendida na Ontologia *Detrend*, para a modelagem da análise de regressão e seus relacionamentos.

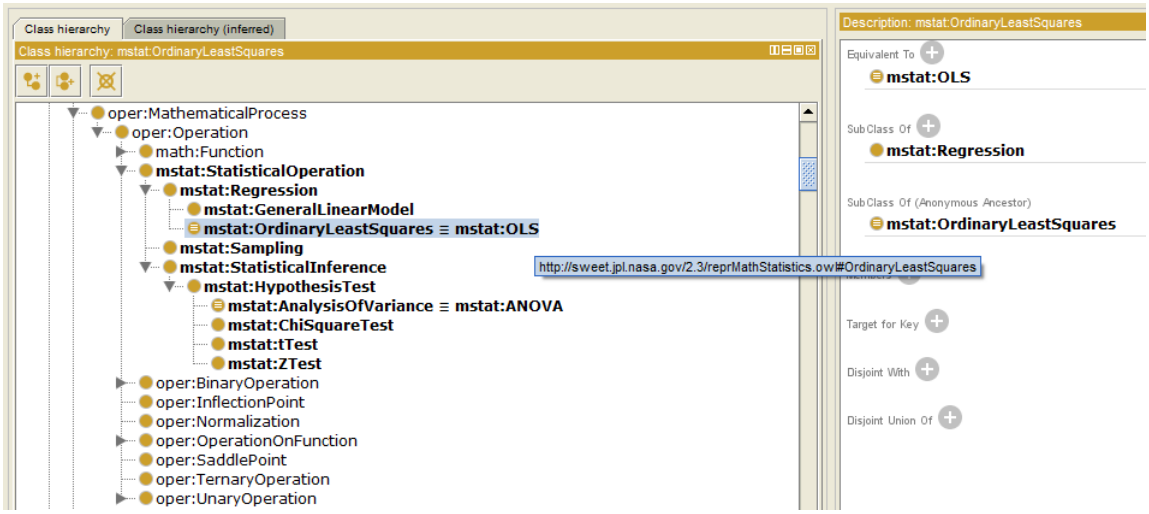


Figura 6.15: Representação da regressão OLS na Ontologia reprMathStatistics.owl

- ***Statistical Analysis*** **[36]**
  A Figura 6.16 mostra a Taxonomia da Ontologia *StatisticalAnalysis*, Versão 1.0, considerada para reuso na Ontologia *Detrend*, por apresentar uma classificação de conceitos detalhada, sendo disponível gratuitamente para *download* em [49]. Essa ontologia estava armazenada em [36], porém, atualmente, a URL não se encontra acessível, mas a ontologia é importada a partir de uma versão salva localmente, onde estão as demais ontologias do modelo.

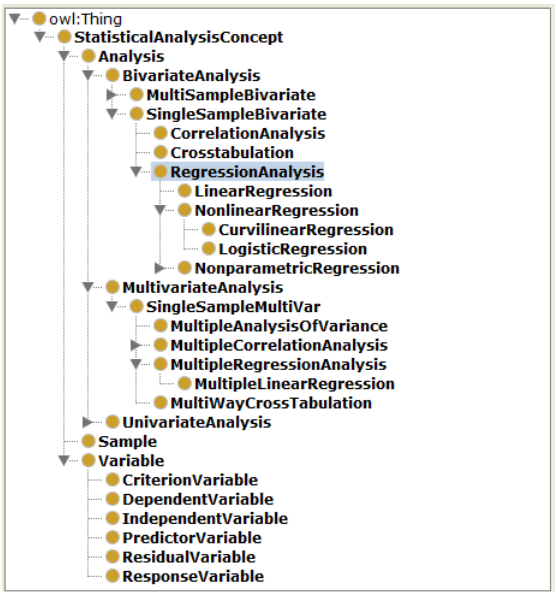


Figura 6.16: Taxonomia da Ontologia *StatisticalAnalysis.owl* reutilizada em *DetrendOntology.owl*.

Essa ontologia (Figura 6.16) foi desenvolvida por Ecoinformatics.org [49], sendo um recurso online para gerenciamento de dados e informações ecológicas, sendo uma solução aberta, desenvolvida a partir de colaboração voluntária de desenvolvedores e pesquisadores, os quais produzem software e sistemas, publicações e serviços, benéficos para as Ciências Ecológicas e Ambientais. A partir do reuso desta ontologia, outros métodos de Análise de Regressão utilizados por algoritmos de *detrending* foram especializados.

5. Integração de ontologia. Desenvolvedores de ontologias devem incluir os recursos ontológicos selecionados por meio das atividades do Cenário 1.

   Como resultado deste Cenário, têm-se uma ontologia implementada, assim como um conjunto de documentos relacionados às diferentes atividades. Na sequência é apresentada a definição da ontologia que modela métodos estatísticos que podem ser usados para extração de tendências em séries temporais e seus respectivos algoritmos e softwares.

6.3.3 Definição da Ontologia DO

Como a ontologia TSO, a ontologia *Detrend* foi projetada para ser desenvolvida em um projeto modular. Da mesma forma, é feito o reuso de declarações a partir do conjunto de

ontologias SWEET. Na ontologia DO, a Ontologia StatisticalAnalysis.owl é importada, a qual é estendida para modelagem dos métodos para correção das séries temporais e sua aplicabilidade em algoritmos de *detrending*. A partir do reuso, determinadas declarações foram especializadas, assim como novas declarações foram especificadas. Para a extensão/especialização da ontologia, foi feita uma busca em glossários, vocabulários, nomenclaturas e normas. Quanto à definição dos métodos estatísticos, os mesmos são definidos a partir de bibliografias de Análise de Séries Temporais. As seguintes referências também foram consultadas e analisadas quanto às definições dos conceitos.

- *Glossary Of Statistical Terms* [13]
- *Terminology on Statistical Metadata* [44]
- *International Statistical Institute* [14]
- *Statistics Glossary* [39]
- *Statistical Techniques in the Data Library: A Tutorial* [38]
- ISO *Norms* (3534-1:2006, 3534-2:2006 e 3534-3:1999)[19]
- Norma Brasileira ABNT NBR ISO 3534-1. 1a. Ed. 2006 - 2010 [18].
- *DCMI Type Vocabulary* [9], a partir do qual é criada e definida a classe dcmitype: Software, sendo estendida para modelar os tipos de softwares.
- ACM *Taxonomy* [43], a partir da qual são criadas as classes e subclasses de (do:GeneralProgrammingLanguages).

A saída do Cenário 8 é a ontologia que representa o domínio esperado, a qual é implementada na linguagem OWL. Na sequência é apresentada a definição das classes, relacionamentos e objetos da ontologia DO.

A partir das questões de competência, as classes e relacionamentos foram identificados, assim como algumas instâncias. Restrições nas classes e relacionamentos são declaradas usando axiomas, adicionando semântica e permitindo inferências.

Todas as classes e propriedades de dados e de objetos são anotadas por meio da *tag* (rdfs:comment), identificando qual é a fonte da definição, assim como por meio da *tag* (rdfs:label) que é usada para rotular os elementos. Isto contribui para o entendimento dos conceitos, assim como permite saber qual a proveniência das definições. A Figura 6.17 apresenta uma consulta quanto ao uso dessas *tags*, com as definições das classes e suas referências.

Para a definição da Ontologia DO, alguns pontos foram analisados, os quais são considerados na modelagem. Na fase de pré-processamento, o pesquisador pode observar os componentes das séries temporais isoladamente, aplicando métodos estatísticos específicos para correção de determinado componente. Caso determinada série temporal apresente o componente ruído (irregular ou aleatório) de forma excessiva, pode ser necessário o uso de um método de filtro para correção deste, processo conhecido como *denoising*, antes da estimação ou extração da tendência. Essa tese considera também esse cenário, onde as séries temporais podem ser corrigidas de ruído. Para tanto, são modelados algoritmos e softwares que podem ser utilizados para esse fim.

SPARQL query:

```
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix owl:<http://www.w3.org/2002/07/owl#>
prefix dc:<http://purl.org/dc/elements/1.1/>
prefix tso:<http://www.semanticweb.org/ontologies/2013/7/TimeSeriesOntology.owl#>
prefix do:<http://www.semanticweb.org/ontologies/2013/7/DetrendOntology.owl#>
prefix dpm:<http://www.semanticweb.org/ontologies/2013/11/DetrendProvenanceModel.owl#>
prefix opmo:<http://openprovenance.org/model/opmo#>
prefix dbpedia:<http://dbpedia.org/page/>

SELECT distinct ?class ?comment ?label
WHERE {
        ?class rdf:type owl:Class ;
                rdfs:comment ?comment ;
                rdfs:label ?label .} ORDER BY ?class
```

class	comment	label
LocalPolynomialRegression	"Definition: The value of the regression function for the point is then obtained by evaluating the local polynomial using the explar	"LocalPolynomialRegressior
LocalRegression	"Definition: In this case occurs the fitting of simple models to localized subsets of the data to build up a function that describes tl	"LocalRegression"^^<http:
LocalRegressionBasedDetrendingAlgorithm	"Definition: This class describes the local regression based detrending algorithm."^^<http://www.w3.org/2001/XMLSchema#stri	"LocalRegressionBasedDetr
LocalRegressionBasedDetrendingSoftware	"Definition: This class describes the local regression based detrending software."^^<http://www.w3.org/2001/XMLSchema#strii	"LocalRegressionBasedDetr
LocalRegressionBasedSmoothing	"Definition: Local regression is an old method for smoothing data, having origins in the graduation of mortality data and the smo	"LocalRegressionBasedSmo
LocallyWeightedLinearRegression	"Definition: If the data contains outliers, the smoothed values can become distorted, and not reflect the behavior of the bulk of tl	"LocallyWeightedLinearReg
LocallyWeightedPolynomialRegression	"Definition: LOESS, originally proposed by Cleveland (1979) and further developed by Cleveland and Devlin (1988), specifically de	"LocallyWeightedPolynomia
LocallyWeightedQuadraticRegression	"Definition: If the data contains outliers, the smoothed values can become distorted, and not reflect the behavior of the bulk of tl	"LocallyWeightedQuadratic
Loess	"Definition: This class represents the Loess method."^^<http://www.w3.org/2001/XMLSchema#string>	"Loess"^^<http://www.w3
LoessRegression	"Definition: LOESS, originally proposed by Cleveland (1979) and further developed by Cleveland and Devlin (1988), specifically de	"LoessRegression"^^<http
LowPassFilter	"Definition: A low pass filter passes the lower or slower frequencies. Source: Shumway, R. H. and Stoffer, D. S. Time Series Analy	"LowPassFilter"^^<http://v
Lowess	"Definition: This class represents the Lowess method."^^<http://www.w3.org/2001/XMLSchema#string>	"Lowess"^^<http://www.v
LowessRegression	"Definition: Use Lowess models to fit smooth surfaces to your data. The names "lowess" and "loess" are derived from the term	"LowessRegression"^^<ht
MaximumLikelihoodEstimationMethod	"Definition: Maximum likelihood estimation begins with writing a mathematical expression known as the Likelihood Function of th	"MaximumLikelihoodEstimat
Mean	"Definition: This class represents the Mean method."^^<http://www.w3.org/2001/XMLSchema#string>	"Mean"^^<http://www.w3
MeanFilter	"Definition: The probably best known linear filter is the mean filter, where all filter weights are uniformly distributed. Source: Meir	"MeanFilter"^^<http://ww
MedianFilter	"Definition: The median filtering is a nonlinear filter. Source: Kim, S. and Koh, K. and Boyd, S. and Gorinevsky, D. l1 Trend Filtering	"MedianFilter"^^<http://ww

Figura 6.17: Definições e rótulos das classes.

Na fase de pré-processamento, um mesmo método pode ser utilizado para mais de uma tarefa, conforme explicado por [58, 193], que citam o caso do método SSA que pode ser utilizado para: detecção e extração de componentes quasi-periódicos, extração de tendências, *denoising*, previsão e detecção de *change-point*. Da mesma forma, outros métodos de filtros podem ser usados, tanto para *detrending* quanto para *denoising*, conforme [128]. Meinl [183] cita que os métodos de suavização (*smoothing*) e *denoising* podem ser usados como sinônimos por alguns autores. Porém, a suavização denota a remoção de detalhes irregulares (por exemplo *sharks´s fins*) e produz uma versão suave das séries temporais originais. *Denoising*, por outro lado, especificamente objetiva a remoção de ruído (variações de curto prazo com baixas amplitudes), a partir das séries temporais, que não necessariamente resulta em um sinal suave [183].

Com o objetivo de modelar esse cenário, na ontologia DO, são descritos algoritmos e softwares de *detrending* e de *denoising*. Algoritmos classificados como de filtro fazem a remoção de ruído das séries temporais, onde não ocorre nenhuma estimação ou remoção de tendência. Algoritmos de *detrending* fazem a estimação e/ou remoção de tendências, a partir das séries temporais. O Diagrama de Classes (Figura 6.18) apresenta as principais classes da ontologia DO.

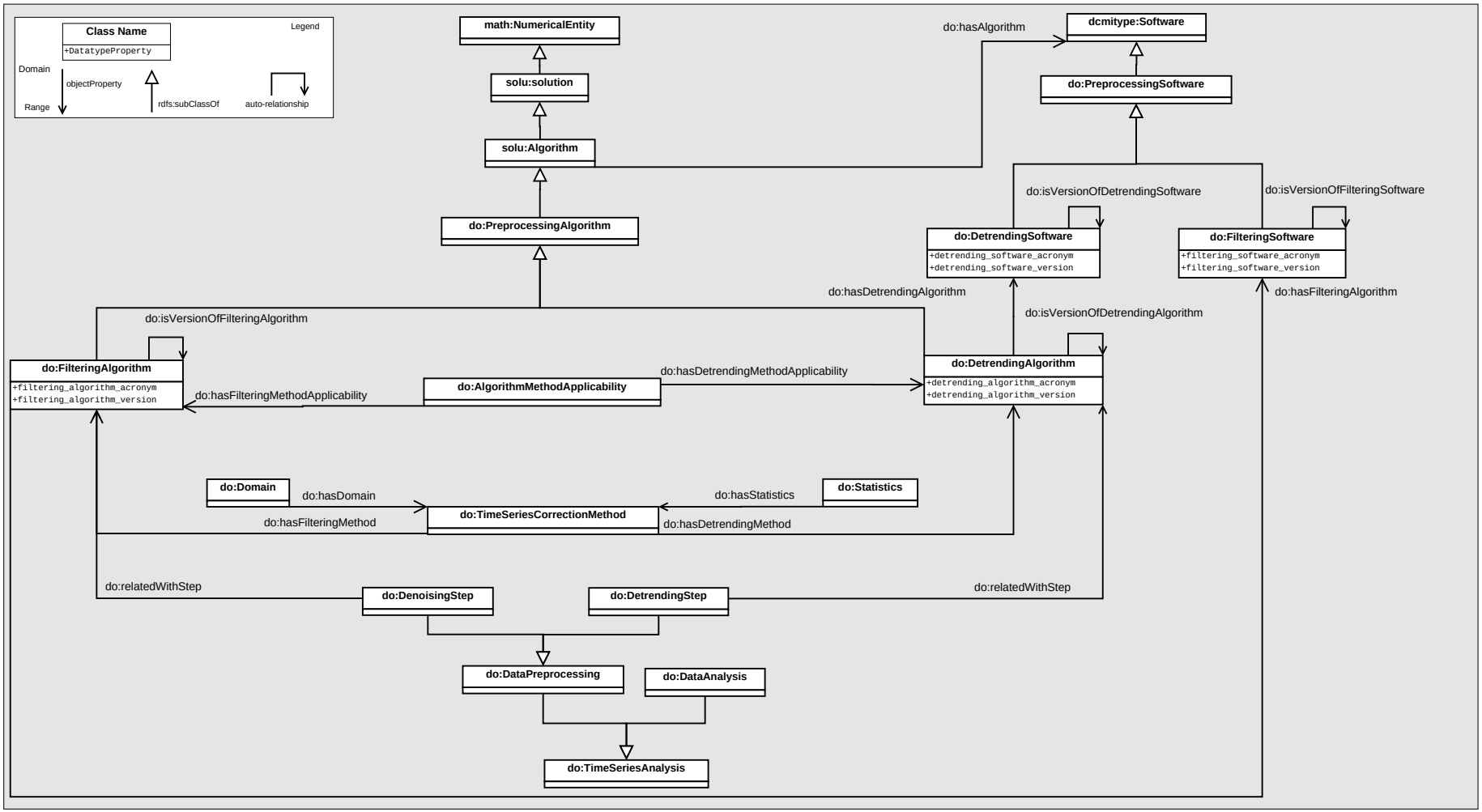


Figura 6.18: Diagrama de Classes da Ontologia DO.

Na Figura 6.18, a partir do reuso de (math:NumericalEntity), subclasse (solu:Solution) e (solu:Algorithm), são criadas as classes (do:PreprocessingAlgorithm) e subclasses (do:FilteringAlgorithm) e (do:DetrendingAlgorithm). A partir de (dcmitype:Software), é criada a subclasse (do:PreprocessingSoftware) e suas subclasses (do:DetrendingSoftware) e (do:FilteringSoftware). Os algoritmos e softwares criados podem ser versões de outros algoritmos ou softwares existentes, modelados conforme um auto-relacionamento na respectiva classe.

Apesar de os algoritmos de *detrending* também realizarem outras atividades do pré-processamento das séries temporais, tais como remoção de *outliers*, agrupamento de séries temporais, remoção de *jumps*, entre outras, os mesmos são classificados na ontologia DO conforme o método utilizado para estimação ou remoção da tendência. Em uma extensão da ontologia, estes podem ser classificados considerando demais atividades.

Na ontologia DO, considerando que um mesmo método pode ser utilizado para realizar mais de uma tarefa nas séries temporais, foram criadas as classes (do:TimeSeriesCorrectionMethod), onde são inseridos métodos e a classe (do:AlgorithmMethodApplicability), para descrever a aplicabilidade do método em determinado algoritmo. Para exemplificar, (do:Moving_Average), referente ao método de médias móveis, pode ser usado em um algoritmo para a tarefa de *denoising*, constituindo, em linguagem de frequências, de um filtro que permite passar o componente da série temporal de baixa frequência como a tendência e remover o componente de alta frequência como o ruído. Esse método também pode ser usado em um algoritmo como uma forma de estimação de tendência não-paramétrica, onde nesse caso particular, o pesquisador considera flutuações (irregularidades) como uma parte intrínseca da tendência, conforme explicado por [95]. Essa escolha depende do contexto das séries temporais, a qual é considerada na modelagem da ontologia, ou seja, é possível declarar o método e qual sua aplicabilidade no respectivo algoritmo. O objetivo da modelagem é facilitar o entendimento sobre o método utilizado e o que foi feito nas séries temporais com sua aplicação.

O Diagrama de Classes da Figura 6.18 descreve o domínio dos métodos, relacionando-os com a classe (do:Domain) contendo as instâncias (do:Time), (do:Time_Frequency) e (do:-Frequency), as quais, assim como outras instâncias, são associadas com a DBpedia como forma de interoperabilidade semântica. A classe (do:Statistics) refere-se à Estatística dos métodos (Paramétrica, Semi-Paramétrica ou Não-Paramétrica).

A modelagem do passo de *denoising* justifica-se pelo fato de que é possível usar os mesmos métodos que podem ser aplicados para extração de tendências, mas com outro enfoque. Na ontologia DO, a partir da classe (do:TimeSeriesAnalysisMethod) e sua subclasse (do:TimeSeriesPreprocessingMethod), é criada a classe (do:TimeSeriesCorrection-Method) e suas subclasses para descrição dos respectivos métodos usados pelos algoritmos (Figura 6.19). A classe (do:RobustMethod) contempla os métodos robustos a *outliers*, onde uma vez declarada para uso uma instância desta classe, sabe-se que o método é resistente à ocorrência de eventos extremos (pontos discrepantes) nas séries temporais. Demais métodos de pré-processamento usados para *detrending* são abordados no decorrer desta seção.

A Figura 6.20 apresenta a aplicabilidade dos métodos nos algoritmos. Um determinado método pode ser usado para estimação de tendência de forma paramétrica, usando um método de análise de regressão ou, de forma não-paramétrica, usando regressão não-paramétrica, a qual é baseada em algum método de suavização [95].

A aplicabilidade do método também pode estar relacionada à remoção da tendência, usando um filtro que permite passar alta frequência, onde a tendência é eliminada das

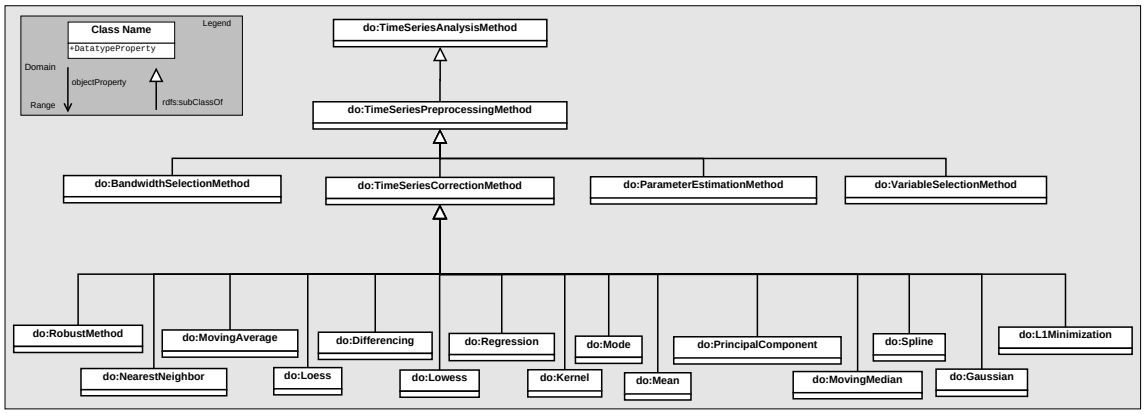


Figura 6.19: Classe (do:TimeSeriesCorrectionMethod) e subclasses.

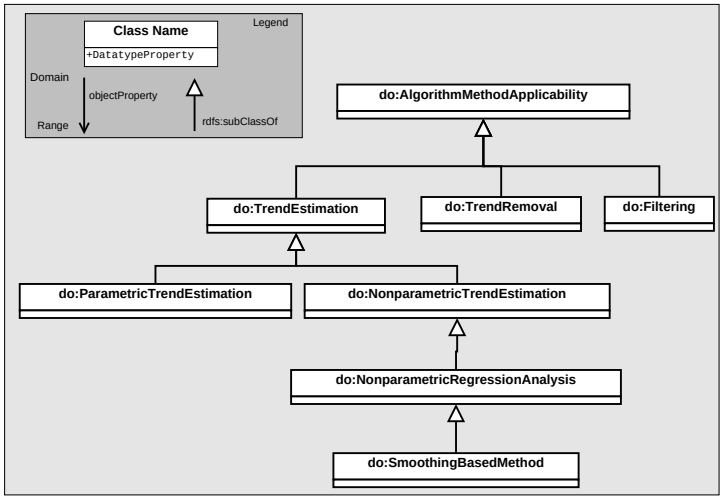


Figura 6.20: Classe (do:AlgorithmMethodApplicability).

séries temporais, deixando passar o componente ruído que é de alta frequência ou, ainda, pode estar relacionada ao uso de filtros para remoção do ruído. A próxima seção apresenta uma extensão da ontologia *Detrend* como forma de demonstrar sua extensibilidade quanto à modelagem de métodos no domínio tempo-frequência.

6.3.4 Extensibilidade da Ontologia DO

A ontologia DO foi desenvolvida a partir do reuso da Ontologia *Statistical Analysis* para métodos de *detrending* no domínio do tempo. Como forma de demonstrar a extensibilidade da ontologia, são modelados alguns métodos de *detrending* com aplicabilidade no domínio tempo-frequência: *Singular Spectrum Analysis* (SSA), proposto por Vautard e Ghil [239], *Empirical Mode Decomposition* (EMD) [156] e, uma variação deste, o *Ensemble Empirical Mode Decomposition* (EEMD) [247].

Esses métodos são classificados como filtros adaptativos aos dados de forma não-linear, os quais podem ser utilizados para extração de tendências das séries temporais. Destaca-se que os mesmos métodos também podem ser usados para extração de ruído, como em [128]. A Figura 6.21 mostra a extensão a partir da classe (do:Filtering). Os métodos EMD e EEMD apresentam a propriedade referente ao número total de funções de modo intrínsecas consideradas (do:total_number_of_IMFs).

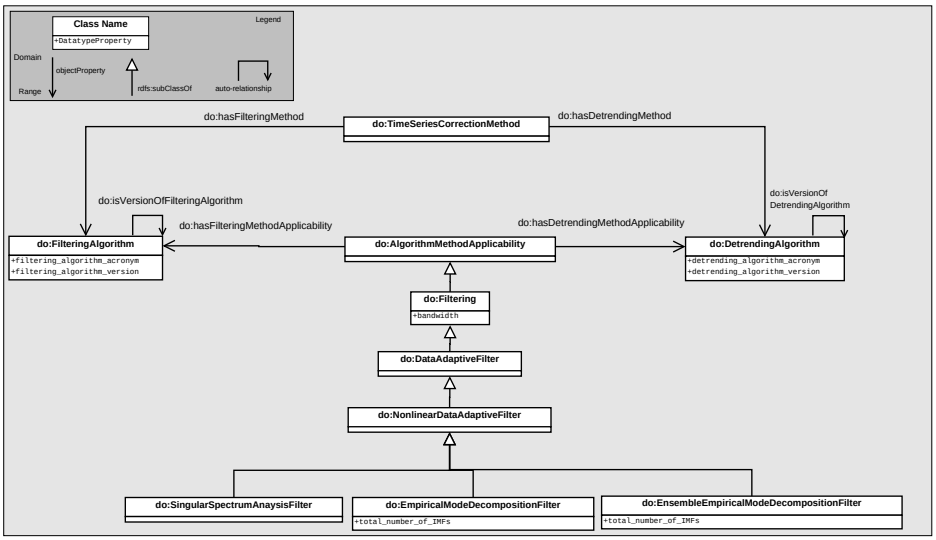


Figura 6.21: Classe (do:DataAdaptiveFilter).

Na Ontologia DO, são criadas classes para representar os respectivos métodos na classe (do:TimeSeriesCorrectionMethod), assim como são criados as respectivas classes de algoritmos e softwares. A partir da classe (do:DetrendingAlgorithm) e de *detrending* baseado em filtro, foram criadas as subclasses de algoritmo de *detrending* baseado em filtro adaptativo aos dados e algoritmo de *detrending* baseado em filtro adaptativo aos dados não-linear. Da mesma forma foram criadas as classes para softwares. A classe (do:TrendRemoval) da Figura 6.22 é estendida para modelar os métodos de *detrending* baseados em filtros adaptativos aos dados (do:DataAdaptiveFilterBasedDetrending), sendo adicionado o axioma (do:hasFilter some do:DataAdaptiveFilter).

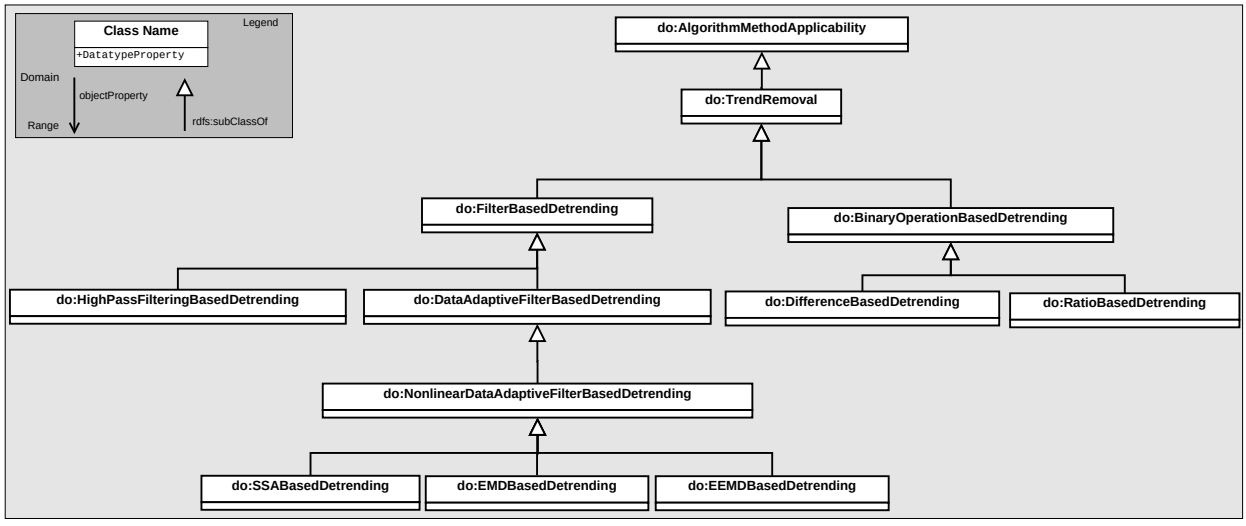


Figura 6.22: Classe (do:TrendRemoval) estendida.

Na ontologia DO, no domínio tempo-frequência, alguns métodos de filtro para estimação de tendência (*trend estimation* ou *trend filtering*) [192] também foram incluídos na ontologia, tais como Hodrick-Prescott [193], l1 *trend filtering* [164], entre outros.

Conforme descrito nesta seção, a ontologia DO permite adicionar conhecimento, quanto aos métodos usados para *detrending* e sua aplicabilidade, assim como para correção das

séries temporais quanto ao ruído.

O Apêndice F apresentada as Figuras E.1 a E.29, referentes aos diagramas dos métodos de *detrending*, como análise de regressão, suavização, filtro e seus parâmetros. Ao final deste Apêndice, encontram-se consultas que respondem às questões de competência, desenvolvidas em linguagem SPARQL (Figuras E.30 a E.52).

A próxima seção descreve a especificação dos requisitos para desenvolvimento da ontologia referente ao Modelo de Proveniência *Detrend*, incluindo os Cenários 1, 3 e 8 relacionados à Metodologia *NeOn*.

6.4 *Detrend Provenance Model* (DPM)

O Cenário 1 da Metodologia *NeOn* contempla a definição dos requisitos da ontologia, conforme abaixo descrito.

6.4.1 Documento de Especificação de Requisitos (DPM ORSD)

O ORSD contempla o propósito, o escopo, a linguagem de implementação, os usuários e os requisitos da ontologia, conforme segue.

1. Propósito: O propósito de desenvolver a ontologia DPM é descrever e inferir conhecimento semântico quanto aos artefatos (séries temporais), processos e agentes de *detrending*. O modelo é desenvolvido com base na literatura de Análise de Séries Temporais.

2. Escopo: O escopo do modelo refere-se ao escopo da ontologia importada DO, a qual modela métodos no domínio do tempo, podendo ser expandida para o domínio da frequência, conforme exemplo apresentado na Seção 6.3.4.

3. Linguagem de Implementação: A ontologia é implementada na linguagem OWL e a linguagem SPARQL é utilizada para desenvolvimento de consultas.

4. Usuários Finais Pretendidos: Os usuários finais são os pesquisadores de análise de séries temporais e proveniência de dados.

5. Usos Pretendidos: Envolvem a inferência e o desenvolvimento consultas ricas semanticamente quanto à geração de informações de proveniência no passo de extração de tendências.

6. Requisitos da Ontologia: Estes dividem-se em não-funcionais e funcionais, conforme:

   - Requisitos Não-Funcionais (RNF):
     RNF1. A terminologia usada na ontologia DPM é baseada na nomenclatura relacionada ao contexto.

     RNF2. A ontologia é definida no idioma Inglês.

     RNF3. A ontologia é escrita seguindo a seguinte convenção de nomenclatura: nomes de classes são criados em maiúscula, sem usar caracteres alfa-numéricos

ou especiais; relacionamentos iniciam com minúscula e as demais palavras com a inicial maiúscula. Sempre que possível, nos relacionamentos, utilizar a convenção (*hasProperty*), por exemplo, (*hasDetrendingSoftware*); instâncias iniciam com maiúsculas e usam *underline* quando envolver palavras conjuntas, por exemplo (*CDA_Graph*).

- Requisitos Funcionais (RF): os requisitos funcionais que envolvem as Questões de Competência da Ontologia DPM são listadas a seguir. A partir das questões de competência, foi extraído um pré-glossário de termos.

Questões de Competência (QCs) - Ontologia DPM

QC1. Questões genéricas

1. Quais artefatos (séries temporais), processos e agentes estão envolvidos no passo de *detrending* do pré-processamento das séries temporais?

   As Questões 2 a 6, relacionam-se aos principais relacionamentos do Modelo de Proveniência OPM, respectivamente:

   - *WasGeneratedby* (qual artefato (dado) foi gerado por qual processo).
   - *Used* (qual dado foi usado por qual processo).
   - *WasControlledBy* (qual processo foi controlado por qual agente).
   - *WasTriggeredBy* (qual processo foi disparado por outro processo).
   - *WasDerivedFrom* (qual dado foi derivado a partir de outro dado).

2. Quais séries temporais foram geradas e corrigidas de tendência por qual processo de *detrending*?

3. Quais séries temporais foram usadas por quais processos de *detrending*?

4. Quais processos de *detrending* foram controlados por um agente de *detrending*?

5. Quais processos de *detrending* foram disparados por outro processo de *detrending*?

6. Quais séries temporais foram derivadas a partir de outras séries temporais?

7. Quais são as informações de proveniência que podem ser obtidas sobre as séries temporais envolvidas em processo de *detrending*?

8. Quais são as informações e parâmetros que podem ser obtidas sobre os métodos envolvidos em um processo de *detrending*?

9. Quais artefatos, processos e agentes estão envolvidos no passo de filtragem para remoção de ruído que ocorre antes de um passo de *detrending*?

6.4.2 Reuso de Recursos Ontológicos

As atividades desenvolvidas para reuso de recursos ontológicos na Ontologia DPM são:

1. Pesquisa de ontologias. Para realizar esta atividade, pesquisas por recursos ontológicos que satisfaçam os requisitos da ontologia foram efetuadas em repositórios e registros tais como Swoogle [53] e Watson [57], assim como utilizando o *Plugin* Watson [56], a partir da ferramenta *NeOn* [233].

   Para definição da ontologia relacionada ao modelo de proveniência, o reuso foi considerado como uma atividade relevante, buscando interoperabilidade semântica. Para seu desenvolvimento, as seguintes ontologias relacionadas a modelos de proveniência foram analisadas quanto à possibilidade de reuso:

   - Linguagem de Marcação de Provas - *Proof Markup Language* (PML)[182]
   - Provenir [220]
   - Modelo de Proveniência Aberto - *Open Provenance Model* (OPM) [197, 194]

2. Avaliação da ontologia. Nessa atividade, os recursos ontológicos obtidos na atividade anterior são inspecionados, quanto ao conteúdo e granularidade, objetivando verificar se tais recursos satisfazem os requisitos de ORSD.

3. Comparação de ontologias. Os recursos ontológicos são comparados conforme custos econômicos, clareza do código e qualidade do conteúdo.

   Quanto às atividades anteriores, as ontologias PML e Provenir não foram consideradas para reutilização na definição da ontologia, inclusive envolvendo especialistas do domínio para esta análise, conforme:

   - A ontologia é relacionada a uma área específica, podendo ser adaptada para o contexto em questão.
   - Existe possibilidade de reuso da ontologia, mas seu uso implica em uma adequação à forma de modelagem.
   - Modelos de proveniência são incentivados a um mapeamento ao Modelo OPM pelo Consórcio W3C como forma de interoperabilidade.

4. Seleção de ontologia. Desenvolvedores devem selecionar o conjunto de recursos ontológicos mais apropriados, baseados nos critérios definidos.

   Dentre as ontologias analisadas para reuso, optou-se pelo uso do modelo OPM, devido às suas características de ser um modelo independente de tecnologia, recomendado pelo Consórcio W3C para reuso quanto à proveniência, sendo uma ontologia adequada ao contexto em questão.

   A definição quanto ao modo de reuso, destaca-se que os recursos ontológicos são reutilizados conforme ontologia OPM original.

5. Integração de ontologia. Desenvolvedores de ontologia devem incluir os recursos ontológicos selecionados por meio das atividades do Cenário 1.

   Após importar as ontologias na ferramenta Protégé 4.1, a combinação das ontologias TSO e DO ao OPM é feita a partir da classe (tso:TimeSeriesData), a qual é declarada como subclasse de (ns:Artifact).

Como resultado deste Cenário, têm-se uma ontologia implementada, assim como um conjunto de documentos relacionados às diferentes atividades. Na sequência é apresentada a definição do Modelo de Proveniência *Detrend*.

6.4.3 Definição do Modelo DPM

Uma das atividades do Cenário 8 é a modularização da ontologia. Nesse caso, essa atividade não foi necessária devido ao seu projeto modular. A partir do reuso da Ontologia OPM, determinados elementos foram estendidos para modelar a proveniência no passo de extração de tendências das séries temporais. A partir do reuso, determinadas declarações foram especializadas, assim como novas declarações foram especificadas. A saída deste Cenário é a ontologia representando o domínio esperado, implementada na linguagem OWL. Na sequência é apresentada a definição das classes, relacionamentos e objetos da ontologia DPM.

O desenvolvimento modular das ontologias de séries temporais e de métodos de *detrending* contribuiu para a combinação destas com a Ontologia do Modelo de Proveniência Aberto - OPM [194]. Modelos de proveniência contribuem para interoperabilidade, constituindo uma forma comum de representar informações. Dentre os modelos de proveniência existentes, OPM se destaca, apresentando como principais características [194]: possibilita que informações de proveniência sejam trocadas entre sistemas, por meio de uma camada de compatibilidade baseada em um modelo de proveniência compartilhado, o qual permite a geração de um grafo de proveniência; possibilita a construção e o compartilhamento de ferramentas que operam com base no modelo; define proveniência de uma maneira precisa, independente de tecnologia; suporta a representação digital de proveniência para qualquer 'coisa' produzida computacionalmente ou não; permite múltiplos níveis de descrição co-existirem; define um conjunto central de regras que identificam inferências válidas para representar proveniência.

A especificação OPM [196] descreve este modelo como um grafo acíclico dirigido definido como um registro de execuções passadas (ou atuais). Constitui um modelo de artefatos no passado, explicando como estes foram derivados, podendo ser ou em tempo passado ou em execução. É um modelo teórico, onde um conjunto de regras são definidas, permitindo inferências em dependências causais. Apesar de ser independente de tempo, é possível incluir informações de tempo no modelo.

Em um grafo de proveniência OPM, existem três tipos de nós principais: Artefato (*Artifact*), Agente (*Agent*) e Processo (*Process*). Cinco tipos de arestas são suportadas (Tabela 6.1): Usado (*Used*), Foi Gerado Por (*WasGeneratedBy* - WGB), Foi Derivado a Partir (*WasDerivedFrom* - WDF), Foi Controlado Por (*WasControlledBy* - WCB) e Foi Disparado Por (*WasTriggeredBy* - WTB).

Artefato é um conceito geral, representando um pedaço imutável de estado, podendo ter uma personificação em um objeto físico, ou uma representação digital em um sistema computacional. Agente é uma entidade contextual, agindo como um catalisador de um processo. Processo refere-se a uma ação ou séries de ações feitas ou causadas por artefatos, resultando em novos artefatos. Nó é uma classe contendo um conjunto de instâncias em um grafo. Nodos podem ser a fonte ou o efeito de arestas. Uma aresta é um relacionamento causal representada por um arco, constituindo uma dependência entre a fonte do arco (o efeito) e o destino do arco (a causa).

Este modelo captura dependências causais entre entidades. Nodos podem ser um

artefato, um processo ou um agente, conectados por arestas diretas. Uma aresta representa uma dependência causal entre sua fonte, denotando o efeito, e seu destino, denotando a causa, expressando as seguintes dependências: um artefato foi gerado por um processo; um processo usou um arterfato; um processo foi controlado por um agente; um artefato foi derivado a partir de outro artefato e um processo foi disparado por outro processo (Tabela 6.1).

Tabela 6.1: Elementos OPM [196]

Aresta:	Efeito	Causa
Used	Process	Artifact
Used*[1]:	Process	Artifact
WasGeneratedBy	Artifact	Process
WasGeneratedBy*	Artifact	Process
WasDerivedFrom	Artifact	Artifact
WasDerivedFrom*	Artifact	Artifact
WasControlledBy	Process	Agent
WasTriggeredBy	Process	Process

A Ontologia OWL OPMO [23] especifica uma serialização RDF do modelo abstrato OPM. Permite expressividade completa dos conceitos OPM e possibilita a realização de inferências. É baseado no Vocabulário OPM - OPMV [24], o qual descreve os conceitos do OPM, mas não permite inferências. OPM e OPMV são diferentes na forma de codificação das informações. OPMV usa propriedades para codificar as arestas OPM, ao passo que OPMO usa classes explícitas, como (opmo:Used).

Moreau [194] apresenta um índice alfabético dos termos OPMO, definidos por classes (conceitos) e por propriedades (relacionamentos, atributos), sendo:

- Classes: Avalue, Account, Annotable, Annotation, Edge, Entity, EventEdge, Node, OPMGraph, OTime, Property, Role, Used, WasControlledBy, WasDerivedFrom, WasGeneratedBy, WasTriggeredBy.

- Propriedades: account, annotation, avalue, cause, causeUsed, causeControlledBy, causeWasDerivedFrom, causeWasGeneratedBy, causeWasTriggeredBy, content, datapropertyAbbreviation, effect, effectInverse, effectUsed, effectUsedInverse, EffectWasControlledBy, effectWasControlledByInverse, effectWasDerivedFrom, effectWasDerivedFromInverse, effectWasGeneratedBy, effectWasGeneratedByInverse, effectWasTriggeredBy, effectWasTriggeredByInverse, encoding, endTime, exactlyAt, hasAccount, hasAgent, hasArtifact, hasConstituent, hasDependency, hasProcess, key, label, noEarlierThan, noLaterThan, pname, profile, property, role, startTime, time, type, usedStar, value, WasDerivedFromStar, WasGeneratedByStar.

Na sequência é apresentado o Modelo de Proveniência *Detrend*, definido de forma modular e centrado no reuso de recursos ontológicos.

6.4.3.1 Modelo de Proveniência para Extração de Tendências em Séries Temporais

A Figura 6.23 mostra o projeto modular, apresentando as seguintes ontologias, as quais fazem reuso de declarações, a partir do conjunto de Ontologias SWEET:

- *TimeSeriesOntology* (prefixo tso:), descreve a representação do conhecimento sobre a proveniência de séries temporais e seus componentes.

- *DetrendOntology* (prefixo do:), a qual importa a ontologia StatisticalAnalysis para modelagem de métodos de estimação de tendência, por meio de análise de regressão paramétrica, onde um modelo é ajustado aos dados. Essa ontologia é estendida, incluindo a descrição de outros métodos estatísticos não-paramétricos que podem ser usados para estimação e remoção de tendências, incluindo métodos de análise de regressão não-paramétrica e, o uso de filtros, descrevendo qual a aplicabilidade dos métodos nos respectivos algoritmos e softwares.

- *DetrendProvenanceModel* (prefixo dpm:), caracteriza o modelo de proveniência quanto às séries temporais e aos métodos de *detrending*, combinando as duas ontologias citadas com o Modelo OPM, permitindo representar as classes *Artifact* (*time series data*), *Process* (*detrending process*) e *Agent* (*detrending software agent*). Considerando que um mesmo método pode ser aplicado para mais de uma tarefa na fase de pré-processamento das séries temporais, o modelo descreve processos e agentes relacionados com a tarefa de correção de ruído de alta frequência das séries temporais, tais como *filtering process* e *filtering software agent*.

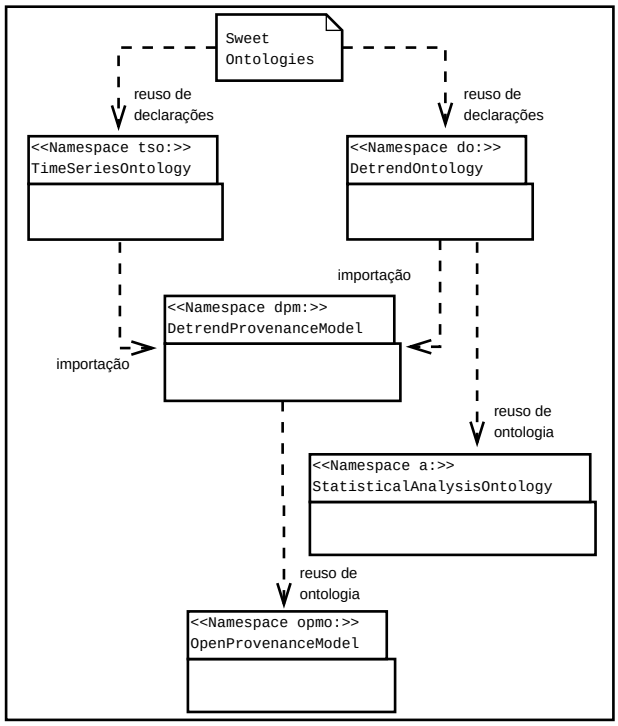


Figura 6.23: Desenvolvimento Modular de Ontologias OWL.

Para o desenvolvimento do modelo, é usado o mecanismo *Imports* da ferramenta Protégé 4.1 para importação da ontologia OPM (prefixo opmo:), a qual importa a ontologia NS (prefixo ns:). Da mesma forma, foram importadas as ontologias de séries temporais (prefixo tso:) e de métodos de *detrending* (prefixo do:), a qual reusa a ontologia StatisticalAnalysis (prefixo a:), conforme a Figura 6.24.

São importados os módulos agregando axiomas, regras e os elementos reutilizados por meio de prefixos a partir das fontes específicas (Figura 6.25).

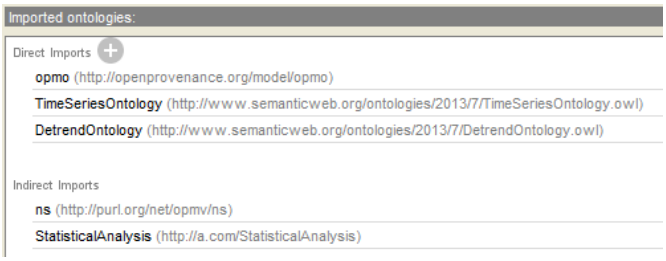


Figura 6.24: Ontologias importadas.

Ontology prefixes:

Prefix	Value
a	http://a.com/StatisticalAnalysis#
dbpedia	http://dbpedia.org/page/
dcmitype	http://purl.org/dc/dcmitype/
do	http://www.semanticweb.org/ontologies/2013/7/DetrendOntology.owl#
dpm	http://www.semanticweb.org/ontologies/2013/11/DetrendProvenanceModel.owl#
openprovenance	http://openprovenance.org/model/#
opmo	http://openprovenance.org/model/opmo#
owl	http://www.w3.org/2002/07/owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
tso	http://www.semanticweb.org/ontologies/2013/7/TimeSeriesOntology.owl#
xsd	http://www.w3.org/2001/XMLSchema#

Figura 6.25: Prefixos das ontologias.

6.4.3.2 Reuso e Extensão de OPM

Na ontologia DPM, os nodos OPM e algumas arestas foram especializadas como forma de geração de informações de proveniência semântica quanto à *detrending*. A Figura 6.26 mostra exemplos de extensões feitas no Modelo OPM. Após a importação dos módulos, a classe (tso:TimeSeriesData) é definida como uma subclasse de (*Artifact*), onde as séries temporais são um tipo de artefato.

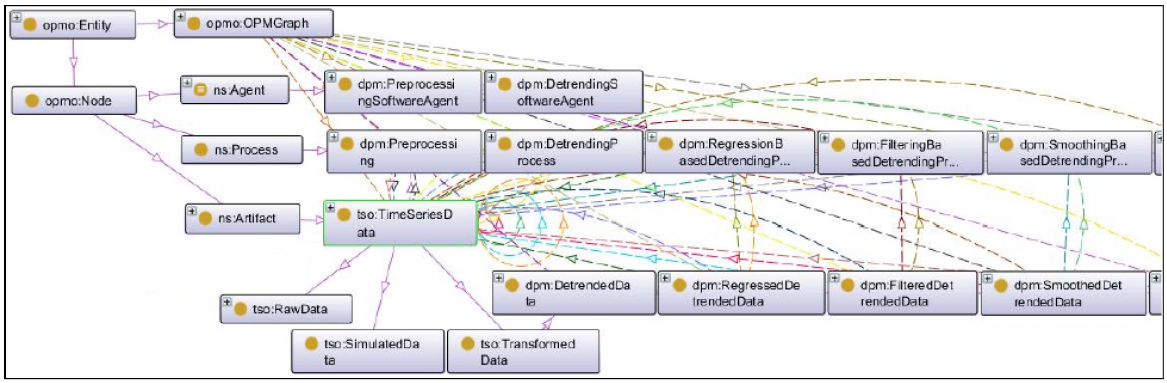


Figura 6.26: Extensões no Modelo OPM.

Independente do método utilizado, ao corrigir uma série temporal de tendência, uma nova série é gerada. Da mesma forma ocorre com a aplicabilidade de métodos de correção de séries temporais para eliminar ou reduzir o ruído, gerando séries filtradas.

Após uma análise na bibliografia de séries temporais sobre o tipo de série que é gerada,

a partir da execução de um determinado processo de *detrending*, a classe (tso:TransformedData) é especializada em (dpm:DetrendedData). As subclasses são estendidas considerando o tipo de dado resultante, a partir do método de estimação ou de remoção de tendência, utilizado para *detrending* (Figura 6.27). Uma referência dessa nomenclatura está em [184], onde existem os termos *difference detrended data* e *ratio detrended data*, relacionados ao tipo de operação binária usada para remoção da tendência, por subtração ou divisão da série temporal, pela tendência estimada, gerando séries estacionárias. Outro exemplo de nomenclatura é encontrada em [121] com os termos *SSA filtered* e *OLS linear filtered data*.

A classe estendida (dpm:DetrendedData) e subclasses abaixo (Figura 6.27) apresenta o axioma (tso:hasTimeSeriesAssumption tso:Stationarity), inferindo na respectiva classe de séries estacionárias (tso:StationaryTimeSeries) as séries corrigidas de tendência.

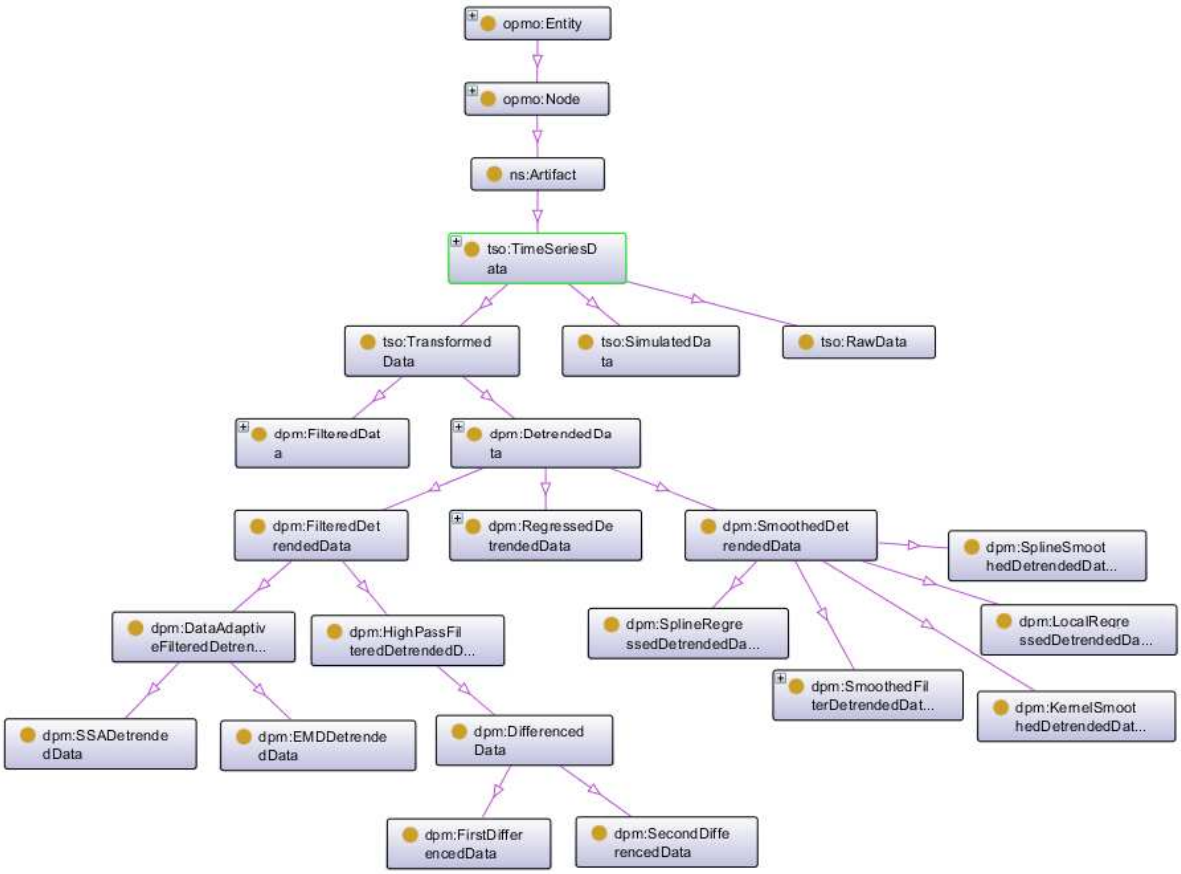


Figura 6.27: Classe (ns:Artifact) estendida.

- (dpm:RegressedDetrendedData): representa as séries temporais corrigidas de tendência onde as mesmas foram estimadas por algum método baseado em regressão paramétrica.
- (dpm:SmoothedDetrendedData): representa as séries temporais corrigidas de tendências onde as mesmas foram identificadas a partir de um método de estimação não-paramétrica, o qual é baseado em alguma forma de suavização. Essa classe é especializada conforme os tipos de métodos de suavização.
- (dpm:SmoothedFilterDetrendedData): representa uma classe especializada da classe anterior, representando as séries temporais corrigidas de tendências, onde as mesmas

foram estimadas por um método não-paramétrico, baseado em algum filtro de suavização, como o filtro de média móvel.

- (dpm:FilteredDetrendedData): representa as séries temporais corrigidas de tendências, onde as mesmas foram filtradas por um método de filtro. Essa classe é especializada nas subclasses a seguir.

- (dpm:HighPassFilteredDetrendedData): representa as séries temporais corrigidas de tendências, onde as mesmas foram filtradas por um método de filtro passa alta frequência. Essa classe é especializada em (dpm:DifferencedData), e subclasses contendo séries extraídas de tendências por um filtro *Differencing*.

- (dpm:DataAdaptiveFilteredDetrendedData): representa as séries temporais corrigidas de tendências, onde as mesmas foram extraídas por um método de filtro adaptativo aos dados. Essa classe é especializada em: (dpm:EMDDetrendedData), contendo séries que foram extraídas de tendências por algum *Empirical Mode Decomposition Trend Filter - EMD Filter* e (dpm:SSADetrendedData), contendo séries extraídas de tendência por um *Singular Spectrum Analysis Filter - SSA Filter*. Essa classe é definida como exemplo de extensibilidade da ontologia para o domínio tempo-frequência.

- (dpm:FilteredData): representa as séries temporais corrigidas de ruído. A partir dos dados filtrados de ruído, um método de estimação ou extração de tendências pode ser aplicado nas séries.

A classe (ns:Process) é estendida para modelar o pré-processamento (dpm:Preprocessing), processos de *detrending* para extração de tendências (dpm:DetrendingProcess) e processos de filtros para extração de ruído (dpm:FilteringProcess).

Essa extensão contribui para a associação quanto ao método utilizado nos processos. Para obter mais informações sobre os métodos e seus parâmetros, a partir do agente que controlou um processo, é possível verificar com qual software o mesmo é associado pela propriedade de objeto (dpm:hasDetrendingSoftware) e, a partir deste, verificar qual o respectivo algoritmo (do:hasDetrendingAlgorithm). A classe (do:DetrendingAlgorithm) tem a propridade de objeto (do:hasDetrendingMethod) e sua aplicabilidade no referido algoritmo de *detrending* (do:hasDetrendingMethodApplicability). As extensões da classe (dpm:DetrendingProcess) incluem:

- (dpm:RegressionBasedDetrendingProcess): representa os processos de *detrending*, relacionados aos métodos de estimação de tendência de forma paramétrica, baseados em algum método de regressão linear, não-linear ou múltipla.

- (dpm:SmoothingBasedDetrendingProcess): representa os processos de *detrending*, relacionados aos métodos de estimação de tendência de forma não-paramétrica, os quais são baseados em alguma forma de suavização.

- (dpm:SmoothingFilterBasedDetrendingProcess): representa os processos de *detrending*, relacionados aos métodos de estimação de tendência de forma não-paramétrica, os quais são baseados em algum filtro de suavização, como o filtro de médias móveis.

- (dpm:FilteringBasedDetrendingProcess): representa os processos de *detrending* relacionados a métodos de *detrending* usando filtros.

- (dpm:HighPassFilterBasedDetrendingProcess): representa os processos de *detrending*, relacionados aos métodos de extração de tendência usando filtros passa alta frequência.

- (dpm:DataAdaptiveFilterBasedDetrendingProcess): representa os processos de *detrending*, relacionados aos métodos de extração de tendência usando filtros adaptativos aos dados, tais como EMD e SSA e suas extensões. Essa classe é definida como exemplo de extensibilidade da ontologia para o domínio tempo-frequência.

- (dpm:FilteringProcess): representa os processos relacionados aos métodos de extração de ruído de alta frequência das séries temporais.

Exceto nos casos da extração de tendências por processos de filtro, como passa alta frequência (*Differencing filter*), a tendência estimada por um método paramétrico ou não-paramétrico, necessita de uma operação binária para sua remoção. No caso de considerar um modelo de decomposição aditivo para a série temporal, a tendência é subtraída a partir das séries temporais. Caso o modelo de decomposição seja considerado como multiplicativo, a tendência é dividida a partir das séries temporais. Isso é declarado na classe de algoritmo de *detrending* baseado em estimação de tendência usando o axioma (dpm:hasTrendRemoval some do:TrendRemovalMethod), onde as instâncias do método de remoção da tendência são associadas com a respectiva operação binária utilizada para *detrending*.

A classe (ns:Agent) é especializada para descrever (dpm:PreprocessingAgent), agente relacionado a um software de pré-processamento, (dpm:DetrendingSoftwareAgent), agente relacionado a um software de *detrending* e (dpm:FilteringSoftwareAgent), agente relacionado a um software de filtro de ruído das séries temporais.

Quanto às arestas (*edges*), estas são especializadas, como forma de se fazer restrições de domínio e intervalo nos relacionamentos de efeito e causa, assim como para adicionar conhecimento semântico (Figura 6.28).

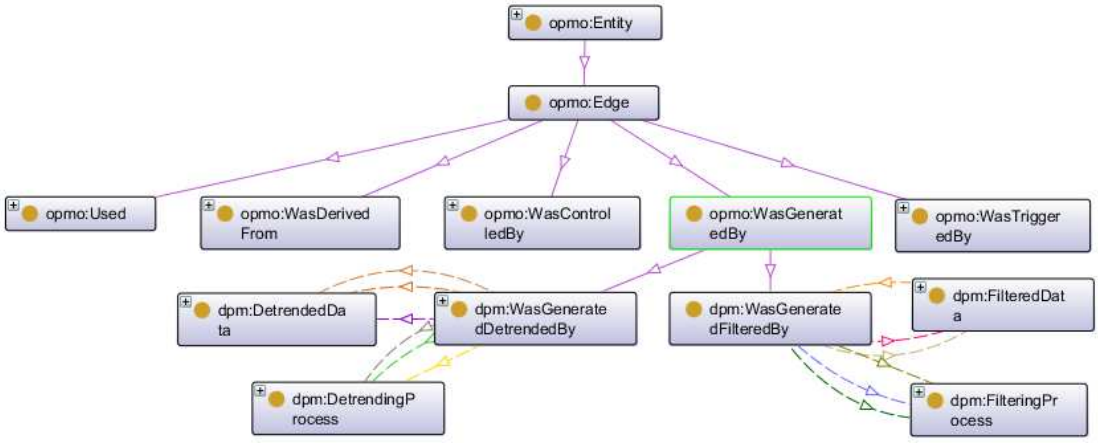


Figura 6.28: Arestas DPM.

A seguir os relacionamentos WGB estendidos são descritos.

- (opmo:WasGeneratedDetrendedBy)): é uma especialização a partir de (opmo:WasGeneratedPreprocessedBy), tendo como efeito ou fonte o domínio (dpm:DetrendedData) e, como causa ou destino, o intervalo referente a (dpm:DetrendingProcess).

- (opmo:WasGeneratedFilteredBy): é uma especialização a partir de (opmo:WasGeneratedPreprocessedBy), tendo como efeito ou fonte o domínio (dpm:FilteredData) e, como causa ou destino, o intervalo referente a (dpm:FilteringProcess).

As demais classes e seus relacionamentos são utilizados conforme definições de domínio e intervalo em OPMO, conforme a Figura 6.29.

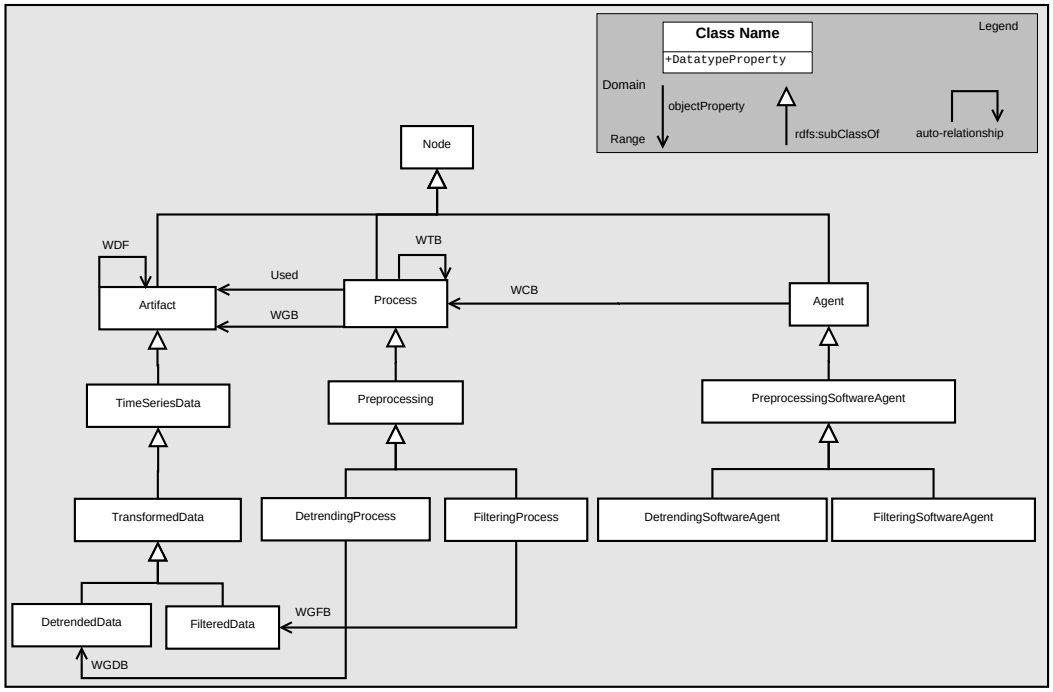


Figura 6.29: Grafo do Modelo de Proveniência *Detrend*.

Da mesma forma que as classes, os respectivos relacionamentos são especializados para restringir o domínio e intervalo na classe WGB:

- (dpm:causeWasGeneratedDetrendedBy), (dpm:effectWasGeneratedDetrendedBy) e (dpm:effectInverseWasGeneratedDetrendedBy), estendidas a partir de (dpm:causeWasGeneratedPreprocessedBy), (dpm:effectWasGeneratedPreprocessedBy) e (dpm:effectInverseWasGeneratedPreprocessedBy), estas especializadas a partir dos respectivos relacionamentos OPM.

- (dpm:causeWasGeneratedFilteredBy), (dpm:effectWasGeneratedFilteredBy) e (dpm:effectInverseWasGeneratedFilteredBy), estendidas a partir de (dpm:causeWasGeneratedPreprocessedBy), (dpm:effectWasGeneratedPreprocessedBy) e (dpm:effectInverseWasGeneratedPreprocessedBy), estas especializadas a partir dos respectivos relacionamentos OPM.

Os demais relacionamentos são utilizados conforme o modelo OPMO. A Tabela 6.2 apresenta as extensões ao Modelo OPM para geração de informações de proveniência na correção de séries temporais.

A Figura 6.30 apresenta o Diagrama de Classes, relacionado à definição da Ontologia DPM, incluindo os relacionamentos de causa e efeito de (WasDerivedFrom), (Used), (WasControlledBy), (WasGeneratedBy) e especializações (WasGeneratedDetrendedBy) e (WasGeneratedFilteredBy), e (WasTriggeredBy).

Tabela 6.2: Extensões ao Modelo OPM.

Edge (Aresta) - Classes	Efeito (fonte) - Domínio	Causa (destinação) - Intervalo
WasGeneratedBy (WGB)	Artifact	Process
WasGeneratedDetrendedBy (WGDB)	DetrendedData	DetrendingProcess
WasGeneratedDetrendedBy (WGDB)	RegressedDetrendedData	RegressionBasedDetrendingProcess
WasGeneratedDetrendedBy (WGDB)	SmoothedDetrendedData	SmoothingBasedDetrendingProcess
WasGeneratedDetrendedBy (WGDB)	SmoothedFilterDetrendedData	SmoothingFilterBasedDetrendingProcess
WasGeneratedDetrendedBy (WGDB)	FilteredDetrendedData	FilteringBasedDetrendingProcess
WasGeneratedDetrendedBy (WGDB)	HighPassFilteredDetrendedData	HighPassFilterBasedDetrendingProcess
WasGeneratedDetrendedBy (WGDB)	DataAdaptiveFilteredDetrendedData	DataAdaptiveFilterBasedBasedDetrendingProcess
WasGeneratedFilteredBy (WGFB)	FilteredData	FilteringProcess
Used	Process	Artifact
Used	DetrendingProcess	TimeSeriesData
Used	RegressionBasedDetrendingProcess	RegressedDetrendedData
Used	SmoothingBasedDetrendingProcess	SmoothedDetrendedData
Used	SmoothingFilterBasedDetrendingProcess	SmoothedFilterDetrendedData
Used	FilteringDetrendingProcess	FilteredDetrendedData
Used	HighPassFilterBasedDetrendingProcess	HighPassFilteredDetrendedData
Used	DataAdaptiveFilterBasedBasedDetrendingProcess	DataAdaptiveFilteredDetrendedData
Used	FilteringProcess	FilteredData
WasControlledBy (WCB)	Process	Agent
WasControlledBy (WCB)	DetrendingProcess	DetrendingSoftwareAgent
WasControlledBy (WCB)	FilteringProcess	FilteringSoftwareAgent
WasTriggeredBy (WTB)	Process	Process
WasTriggeredBy (WTB)	DetrendingProcess	Preprocessing
WasTriggeredBy (WTB)	FilteringProcess	Preprocessing
WasDerivedFrom (WDF)	Artifact	Artifact
WasDerivedFrom (WDF)	DetrendedData	TimeSeriesData
WasDerivedFrom (WDF)	RegressedDetrendedData	TimeSeriesData
WasDerivedFrom (WDF)	SmoothedDetrendedData	TimeSeriesData
WasDerivedFrom (WDF)	SmoothedFilterDetrendedData	TimeSeriesData
WasDerivedFrom (WDF)	FilteredDetrendedData	TimeSeriesData
WasDerivedFrom (WDF)	HighPassFilteredDetrendedData	TimeSeriesData
WasDerivedFrom (WDF)	DataAdaptiveFilteredDetrendedData	TimeSeriesData
WasDerivedFrom (WDF)	FilteredData	TimeSeriesData

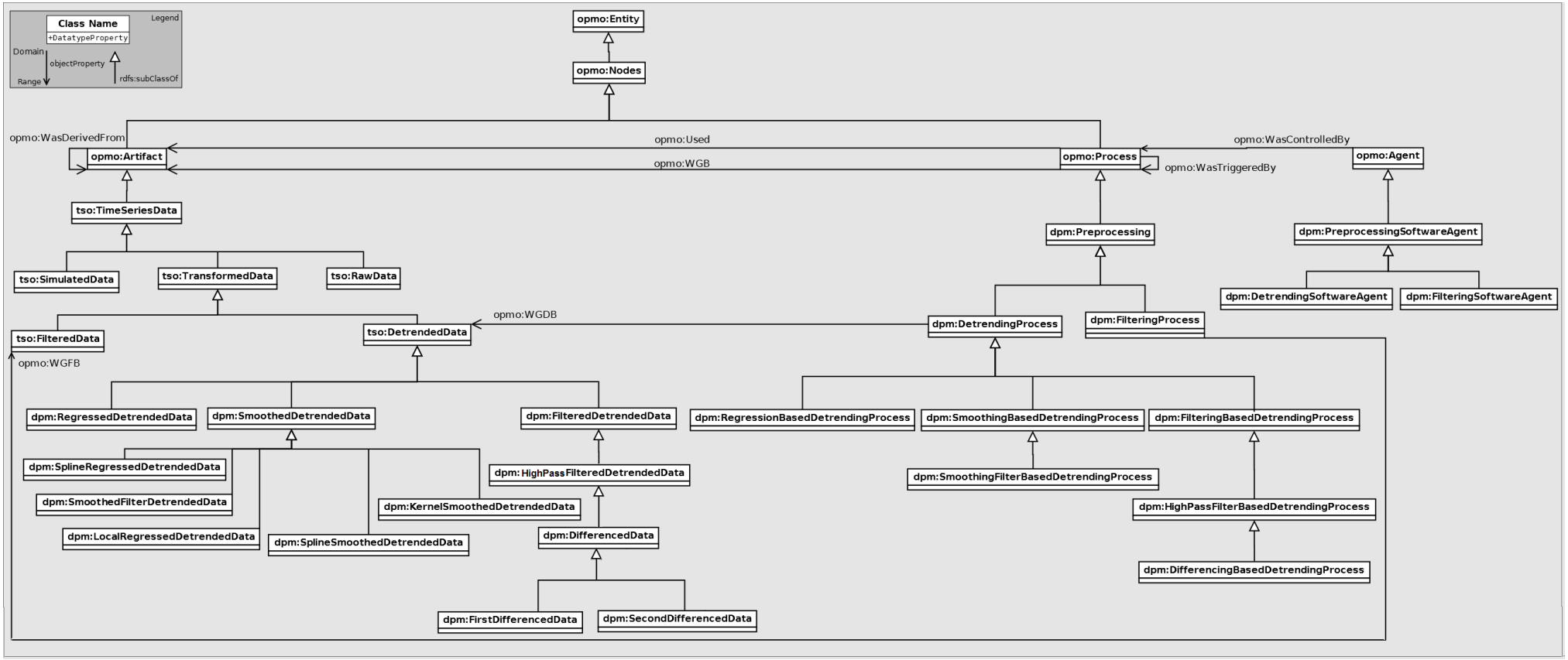


Figura 6.30: Relacionamentos Used, WDF, WGB, WGDB, WGFB e WTB.

A consulta da Figura 6.31 mostra um exemplo de séries temporais corrigidas de tendência a partir de séries filtradas de ruído, apresentando informações sobre as séries originais, qual agente disparou o processo de filtro, qual software a algoritmo são relacionados, assim como o método usado pelo algoritmo e sua aplicabilidade. Nesse caso, após a filtragem do ruído de alta frequência, é possível obter informações sobre a regressão, com o ajuste de um modelo para a tendência de forma paramétrica.

SPARQL query:

```
SELECT   distinct ?rawdata ?observationtype ?observationinterval ?filteringprocess ?filteringagent ?filteringsoftware ?filteringalgorithm ?filteringmethod ?filteringmethodapplicability ?filtereddata ?detrendeddata ?detredingprocess
WHERE {   ?filteringprocess rdf:type dpm:FilteringProcess . ?filtereddata rdf:type dpm:FilteredData .
          ?used1 opmo:causeUsed ?rawdata ;              opmo:effectUsed ?filteringprocess .
          ?wcb1 opmo:causeWasControlledBy ?filteringagent ;
                opmo:effectWasControlledBy ?filteringprocess .
          ?filteringagent dpm:hasFilteringSoftware ?filteringsoftware .
          ?filteringsoftware do:hasFilteringAlgorithm ?filteringalgorithm .
          ?filteringalgorithm do:hasFilteringMethod ?filteringmethod ;
             do:hasFilteringMethodApplicability ?filteringmethodapplicability .
          ?filteringmethodapplicabilitytype rdf:type ?x .
          ?wdf1 opmo:causeWasDerivedFrom ?rawdata ;
                opmo:effectWasDerivedFrom ?filtereddata .
          ?used2 opmo:causeUsed ?filtereddata ;
                 opmo:effectUsed ?detredingprocess .
          ?wdf2 opmo:causeWasDerivedFrom ?filtereddata ;
                opmo:effectWasDerivedFrom ?detrendeddata .
          ?rawdata rdf:type ?rawdatatype ;
                   tso:hasObservationType ?observationtype ;
                   tso:observation_interval ?observationinterval .       ?rawdatatype rdf:type ?x .
          ?detrendeddata rdf:type ?detrendeddatatype .                  ?detrendeddatatype rdf:type ?y .} ORDER BY ?rawdata
```

rawdata	observationtype	observationi...	filteringprocess	filteringagent	filteringsoftware	filteringalgorithm	filteringmethod	filteringmethodapplicability	filtereddata	detrendeddata	detredingprocess
15	Regularly_Sp	"512 sec	Filtering_Proc	Filtering_Software_Ag	Linear_Filtering_Soft	Linear_Filtering_Alg	Single_Moving_Average	Moving_Average_Filter	150	1500	Regression_Bas
16	Regularly_Sp	"512 sec	Filtering_Proc	Filtering_Software_Ag	Linear_Filtering_Soft	Linear_Filtering_Alg	Single_Moving_Average	Moving_Average_Filter	160	1600	Regression_Bas
17	Regularly_Sp	"512 sec	Filtering_Proc	Filtering_Software_Ag	Linear_Filtering_Soft	Linear_Filtering_Alg	Single_Moving_Average	Moving_Average_Filter	170	1700	Regression_Bas

Figura 6.31: Séries temporais corrigidas de tendência a partir de séries filtradas de ruído e informações das séries originais e qual agente, software, algoritmo, método e aplicabilidade do filtro são relacionados.

Dessa forma, no modelo DPM, é possível obter informações de proveniência sobre as séries temporais originais, sobre as séries corrigidas, assim como obter informações quanto aos relacionamentos de domínio e intervalo OPM e sobre os métodos, parâmetros e sua aplicabilidade nos respectivos algoritmos e softwares, gerando conhecimento semântico sobre a correção das séries temporais.

O próximo capítulo apresenta a validação do modelo definido nesta tese, incluindo avaliações individuais por especialistas, o desenvolvimento de um estudo de caso contendo dois casos de uso reais usando arquivos de séries temporais fotométricas e uma análise comparativa desta pesquisa com trabalhos correlatos.

CAPÍTULO 7

VALIDAÇÃO DO MODELO DE PROVENIÊNCIA

Este capítulo apresenta a validação do modelo de proveniência, considerando a Avaliação Funcional da Metodologia NeOn [233], onde as ontologias foram avaliadas em seu contexto de uso, seguido do desenvolvimento de um estudo de caso e uma análise comparativa com trabalhos correlatos.

7.1 Avaliação por Especialistas

A avaliação das ontologias por especialistas é descrita nas seções seguintes. Os Apêndices A, B e C apresentam o documentos utilizados nas avaliações, incluindo um Termo de Compromisso e o Parecer dos Avaliadores. O Apêndice D foi utilizado para verificação do perfil dos avaliadores.

7.1.1 Avaliação da Ontologia TSO

A Ontologia TSO foi avaliada por doze especialistas, destes dois são ontologistas, dois pesquisam sobre ontologias e séries temporais e oito trabalham ou desenvolvem pesquisas com séries temporais. Abaixo segue a análise quanto às respostas obtidas, citando como foram tratadas as sugestões feitas pelos avaliadores.

- A modelagem de séries temporais envolvendo vários intervalos de observação, como por exemplo, uma série medida semanalmente e que passa a ser medida diariamente, é incluída como uma sugestão de extensão na ontologia.
- Quanto à elaboração de questões de competência mais específicas, cita-se que estas foram agrupadas e podem ser resolvidas por meio de consultas na ontologia.
- Em relação à extensibilidade, a ontologia foi desenvolvida para que a mesma possa ser estendida com novas classes, relacionamentos e instâncias, inclusive para domínios específicos.
- Quanto às restrições de domínio e intervalo, estas são contempladas na ontologia sempre que possível.
- O reuso foi considerado desde o início de desenvolvimento da ontologia.
- Quanto à definição das instâncias, estas inicialmente estavam sendo comentadas. Por sugestão de um avaliador, passou-se a comentar somente as classes e relacionamentos.
- Conforme observado por um avaliador, Sazonalidade e Periodicidade passaram a ser tratadas como o mesmo termo. Da mesma forma, considerando a sugestão de outro avaliador, é utilizado medida estatística para média e variância.

- As bibliografias sugeridas por alguns avaliadores foram analisadas, sendo selecionadas para uso as bibliografias Bendat e Pierson [70] e Shumway e Stoffer [223]. Demais fontes citadas nas avaliações que contemplam métodos no domínio da frequência ou métodos a serem estendidos, serão aproveitadas como sugestões para uma extensão da ontologia.

- A sugestão de um avaliador quanto a abordar a não-estacionariedade na co-variância foi incluída na ontologia.

- Um avaliador sugeriu verificar a semântica da *Ontology of Astronomical Object Types* (IVOA) [93], onde a mesma foi analisada e as sugestões de criação da ontologia foram consideradas, porém, esta ontologia modela produtos científicos e não contempla a descrição de séries temporais.

7.1.2 Avaliação da Ontologia DO

Quando a ontologia TSO foi avaliada, a Ontologia DO passou por uma primeira avaliação, contendo métodos paramétricos de estimação de tendência. Após sua extensão para métodos não-paramétricos e o uso de filtros, a Ontologia *Detrend* foi avaliada quanto aos métodos estendidos.

O processo de avaliação ocorreu da mesma forma que na avaliação da ontologia TSO, envolvendo dez avaliadores, sendo os mesmos participantes da avaliação de TSO, os quais receberam os documentos da avaliação (Apêndice B) e uma explicação sobre como os métodos foram modelados na ontologia. Os resultados de ambas as avaliações foram consideradas, destacando-se:

- Além de consultas às bibliografias da área de séries temporais, especialistas contribuíram para a definição dos métodos estatísticos que podem ser usados para *detrending*, dando sugestões de inclusão do projeto (*design*) do filtro, conforme a linguagem de frequências, visto que o *design* do método está relacionado com a passagem ou bloqueio de tendências pelos métodos de *detrending*.

- Outra sugestão é quanto à criação das classes (do:AlgorithmMethodApplicability) e (do:TimeSeriesCorrectionMethod), pois conforme verificado por análises quanto à aplicabilidade dos métodos e também em bibliografias, um mesmo método pode ser usado para realizar mais de uma tarefa na análise de séries temporais. Essa questão é dependente de qual componente está sendo analisando e qual o objetivo da análise, conforme o contexto.

- Quanto à análise de um avaliador que as questões de competência apresentavam questões genéricas juntamente com questões mais específicas, estas foram divididas em categorias para facilitar a interpretação das mesmas.

- Conforme sugestão de um avaliador, filtros não-lineares foram estendidos na ontologia.

- Outra questão discutida com avaliadores é quanto aos métodos de regressão não-paramétrica e o uso de filtros. Por um lado, ao se extrair um componente das séries temporais, a mesma está sendo filtrada. Por outro lado, na ontologia, considerando que nem todos as referências da área descrevem dessa forma, a modelagem considera métodos de suavização como estimação de tendência não-paramétrica.

- Conforme discutido com avaliadores, na presente ontologia, é possível inserir informações conforme a aplicabilidade do método para *detrending*.
- O uso das classes no plural para modelagem das linguagens de programação foi discutido com ontologistas, os quais sugerem deixar dessa forma, dado o embasamento na Taxonomia ACM.

7.1.3 Avaliação do Modelo DPM

Devido ao desenvolvimento modular, foi possível aplicar avaliações individuais nos módulos, onde a participação dos especialistas é considerada essencial. Da mesma forma que a ontologia de *detrending*, a ontologia DPM foi avaliada pelos mesmos dez avaliadores, relacionados com as áreas de séries temporais e desenvolvimento de ontologias. As sugestões dos avaliadores foram consideradas no modelo. O reuso das ontologias TSO e DO no modelo DPM foi avaliado por meio da documentação online das ontologias. Uma das sugestões é quanto às relações de causa e efeito dos nodos OPM, sendo sugerido que as mesmas fossem especializadas nos casos em que contribuíssem para restringir o domínio e o intervalo das relações.

Como forma de validação do modelo, a próxima seção apresenta um estudo de caso para geração de informações de proveniência quanto a métodos aplicados na correção de tendências em séries temporais fotométricas reais, seguido da descrição e análise de trabalhos correlatos.

7.2 Estudo de Caso

Esta seção apresenta um estudo de caso referente ao Modelo de Proveniência *Detrend*, com a geração de informações de proveniência em séries temporais fotométricas reais, as quais são armazenadas em arquivos do tipo FITS (*Flexible Image Transport System*)[146].

Esses arquivos contém curvas de luz, constituindo séries temporais, obtidas geralmente durante intervalos regulares de tempo, a partir de telescópios espaciais, como o *Convenction, Rotation and Planetary Transits* - CoRoT [6], o qual possui dois campos de atuação, detecção e estudo de oscilações estelares (Sismologia Estelar) e procura de planetas extra-solares (Pesquisa de Exoplanetas), sendo este último relacionado ao estudo de caso. Imagens capturadas são disponibilizadas publicamente após o período de um ano de sua obtenção no Archive CoRoT [8].

Para a análise dessas séries temporais, são necessárias, na fase de pré-processamento, correções de fenômenos que ocorrem ao longo do tempo, tais como a extração de tendências (nesse caso a tendência é considerada como um efeito sistemático) e de eventos extremos, assim como ocorre em outras áreas do conhecimento. Diferentes comunidades científicas podem aplicar diferentes algoritmos de *detrend* para os mesmos dados, onde algoritmos de *detrend* removem sinais instrumentais, melhorando o processo de análise.

Trilhar a origem da informação e como os dados foram derivados, na fase de pré-processamento dos dados, é essencial para permitir reuso, reprocessamento e análise nos dados. Esta pesquisa objetiva adicionar conhecimento a este processo, com semântica e padronização, contribuindo para a geração de consultas ricas semanticamente. É apresentado um estudo de caso referente ao modelo definido, a fim de gerar informações de proveniência sobre as séries temporais e os métodos usados para *detrending*, para uso por humanos e/ou agentes de software.

Informações de proveniência podem ser armazenadas no cabeçalho (*header*) dos arquivos do tipo FITS. Porém, a especificação FITS não contempla a adição de metadados de proveniência, descrevendo o uso do metadado HISTORY para armazenar passos executados. Essa forma de geração de proveniência é texto livre, dificultando sua legibilidade por máquina, impedindo seu uso por agentes de software.

Metadados armazenados no *header* das imagens ou em bancos de dados são úteis para pesquisadores locais, mas insuficientes para permitir reuso e reprocessamento pela comunidade científica. Existe uma necessidade de padronização dos metadados de proveniência a serem gerados e armazenados [250], assim como informações mais detalhadas para contemplar as reais necessidades quanto ao conhecimento semântico a respeito da geração de dados, ao longo do tempo. A Figura 7.1 mostra à esquerda uma série temporal apresentando tendências, assim como outros eventos, tais como *jumps* (antes de aplicar um algoritmo de *detrend*) e, à direita, uma nova série temporal, a qual foi gerada e corrigida de tendência por um processo de *detrending*, com a aplicabilidade de um software de *detrending*).

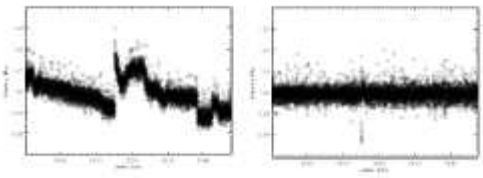


Figura 7.1: *Detrend* em arquivos FITS [190].

A Tabela 7.1 apresenta alguns algoritmos de *detrend*, utilizados para correção de séries temporais de arquivos do tipo FITS.

Tabela 7.1: Algoritmos de *detrending* aplicados em arquivos FITS.

Algoritmo	**Efeito Sistemático**	**Método utilizado**
CoRoT Detrend Algorithm **[190]**	**tendências**	**polinômio do 3º grau**
CoRoT Detrend Algorithm Modified **[78]**	**tendências**	**médias móveis**
SysRem Detrending Algorithm **[181]**	**quaisquer efeitos sistemáticos lineares**	**ajuste linear**
SARS Algorithm **[206]**	**efeito sistemático aditivo**	**método dos mínimos quadrados**
DSTL Algorithm **[189]**	**tendências**	**correlação linear e ajuste polinomial**
TFA Algorithm **[167]**	**tendências**	**filtro linear otimizado pelo método dos mínimos quadrados**

Como forma de demonstração do modelo, são considerados dois casos de uso, referentes à aplicabilidade de métodos para correção de tendências em séries temporais, as quais são analisadas no domínio unidimensional. Os dados podem ser monocromáticos ou estar nos três filtros *Red-Green-Blue* (RGB). Para análises, a variável dependente Y corresponde ao fluxo de observações e a variável independente X corresponde ao tempo, geralmente medido em *Heliocentric Julian Date* (HJD). Independente do método utilizado para *detrending*, uma nova série corrigida (*detrended data*) é gerada.

- Caso de Uso 1. *Detrending* usando regressão polinomial

Este caso de uso é referente à aplicabilidade do algoritmo *Corot Detrend Algorithm* (CDA) [190] nas séries temporais CoRoT. Segundo Mislis et al [190], existem dois problemas instrumentais básicos com todas as curvas de luz CoRoT: i. existe uma tendência de longo prazo, onde a causa física de sua ocorrência não é explicada e; ii. existe a ocorrência de *jumps* nas curvas de luz, onde a maioria das ocorrências é de natureza instrumental.

Algumas suposições são feitas quanto aos dados de séries temporais antes de aplicar o algoritmo CDA, tais como tendências aparecem em quase todas as curvas de luz CoRoT e os fluxos crescente e decrescente podem ocorrer; tendências são não-periódicas, sendo assumido que são um fenômeno de longo prazo; *jumps* são um fenômeno aleatório aparecendo em diferentes filtros em tempos diferentes, observados por uma inspeção visual nos dados. Quanto ao intervalo de observação das séries temporais, segundo Mazeh et al [180], algumas curvas de luz são amostradas com uma medida de 32 segundos, enquanto a maioria é medida com 512 segundos.

O algoritmo CDA, após remover todos os pontos de dados setados como *bad points*, ajusta para todas as curvas de luz um polinômio de terceiro grau para remover a tendência em cada filtro por estrela. Cada curva de luz tem milhares de pontos de dados, nesse caso, o polinômio não ajusta variações de curto prazo e eventos reais de curto prazo como trânsitos planetários[1]. O fluxo é descrito como: Fluxo = a + b.JD + c.JD^2 + d.JD^3, onde JD é a data Juliana normalizada no intervalo (-1 a 1) e (a, b, c e d) são os parâmetros de ajuste para o terceiro grau polinomial. Ao final desse processo, tem-se uma curva de luz *detrended* por filtro para cada estrela. Após esse procedimento, CDA procede para remover os *jumps*.

Nesse estudo de caso, a partir de uma amostra do *short run* SRa01[2], é aplicado o algoritmo CDA, conforme a Figura 7.2. Para sua realização, são consideradas as suposições descritas em [190], assim como com base no conhecimento de pesquisadores da área. As séries temporais da amostra são inseridas na base de conhecimento da Ontologia TSO.

Devido ao desenvolvimento modular e a extensão do modelo OPM, é possível fazer consultas e obter informações a respeito da proveniência das séries temporais armazenadas nos arquivos do tipo FITS, assim como é possível consultar sobre os métodos de *detrending* e seus parâmetros, a partir dos principais nodos OPM (artefato, processo e agente) e de seus cinco principais relacionamentos: (*Used*), WDF, WCB, WTB e WGB, e as extensões definidas, Foi Gerado *Detrended* Por - (WasGeneratedDetrendedBy - WGDB) e, no caso de remoção de ruído, Foi Gerado e Filtrado Por - (WasGeneratedFilteredBy - WGFB).

A extensão foi feita com base no próprio OPM que estende, por exemplo, a relação (cause) em (causeUsed), (causeWasDerivedFrom); e, (effect), especializado em (effectUsed), (effectWasDerivedFrom), entre outras. Também a relação (hasConstituent) do grafo de proveniência (domínio OPMGraph) é estendida com as relações apresentando os domínios *Artifact*, *Agent* e *Process*, respectivamente para (hasArtifact), (hasAgent) e (hasProcess).

Na ontologia DPM são estendidos os relacionamentos (hasPreprocessingAgent) e - (hasDetrendingSoftwareAgent) como subpropriedade de (hasAgent), (hasPreprocessing) e (hasDetrendingProcess) como subpropriedades de (hasProcess) e (hasDetrendedData) como subpropriedade de (hasArtifact). Da mesma forma são estendidos os relacionamentos

[1]Trânsito é um método de busca por exoplanetas nas curvas de luz.

[2]SRA01 significa curta execução do CoRoT, a letra (a) significa direção anti-centro. URL:http://idoc-corotn2-public.ias.u-psud.fr/jsp/CorotFullDownload.jsp, Acesso Fev/2013

para filtros. Para justificar a extensão, além da importância da geração de conhecimento semântico quanto ao passo de *detrending*, existe a necessidade de incluir relacionamentos específicos quanto ao domínio e intervalo dos relacionamentos OPMO, tais como o fato de uma dada série temporal ser gerada e *detrended* somente por um processo de *detrending*, objetivando extração da tendência e não por um outro processo do pré-processamento, como um processo de extração de ruído.

Na sequência são apresentadas as consultas desenvolvidas, a partir deste Caso de Uso, onde pesquisadores podem obter informações sobre as séries temporais, contribuindo na tomada de decisão quanto à execução de processos de *detrending*.

7.2.1 Consultas Relacionadas ao Caso de Uso 1

Com as ontologias integradas e estendidas, é possível fazer consultas envolvendo a proveniência das séries temporais. Considerando os algoritmos e softwares de *detrending* e os relacionamentos, a partir da extensão dos principais nodos OPM, é possível gerar informações de proveniência quanto aos dados e aos métodos e sua aplicabilidade pelos algoritmos.

Na sequência são apresentadas as Figuras 7.2 e 7.3. A Figura 7.2 apresenta as medidas de média e desvio-padrão da série temporal original de identificação (ID) (22) na parte superior, a qual apresenta tendência e, na parte inferior, a nova série gerada e *detrended*, à esquerda, por um ajuste polinomial, como é o caso do algoritmo CDA e, à direita, um ajuste de média móvel que é descrita no Caso de Uso 2. A Figura 7.3 apresenta o histograma dessa série, a qual apresenta uma Distribuição Normal.

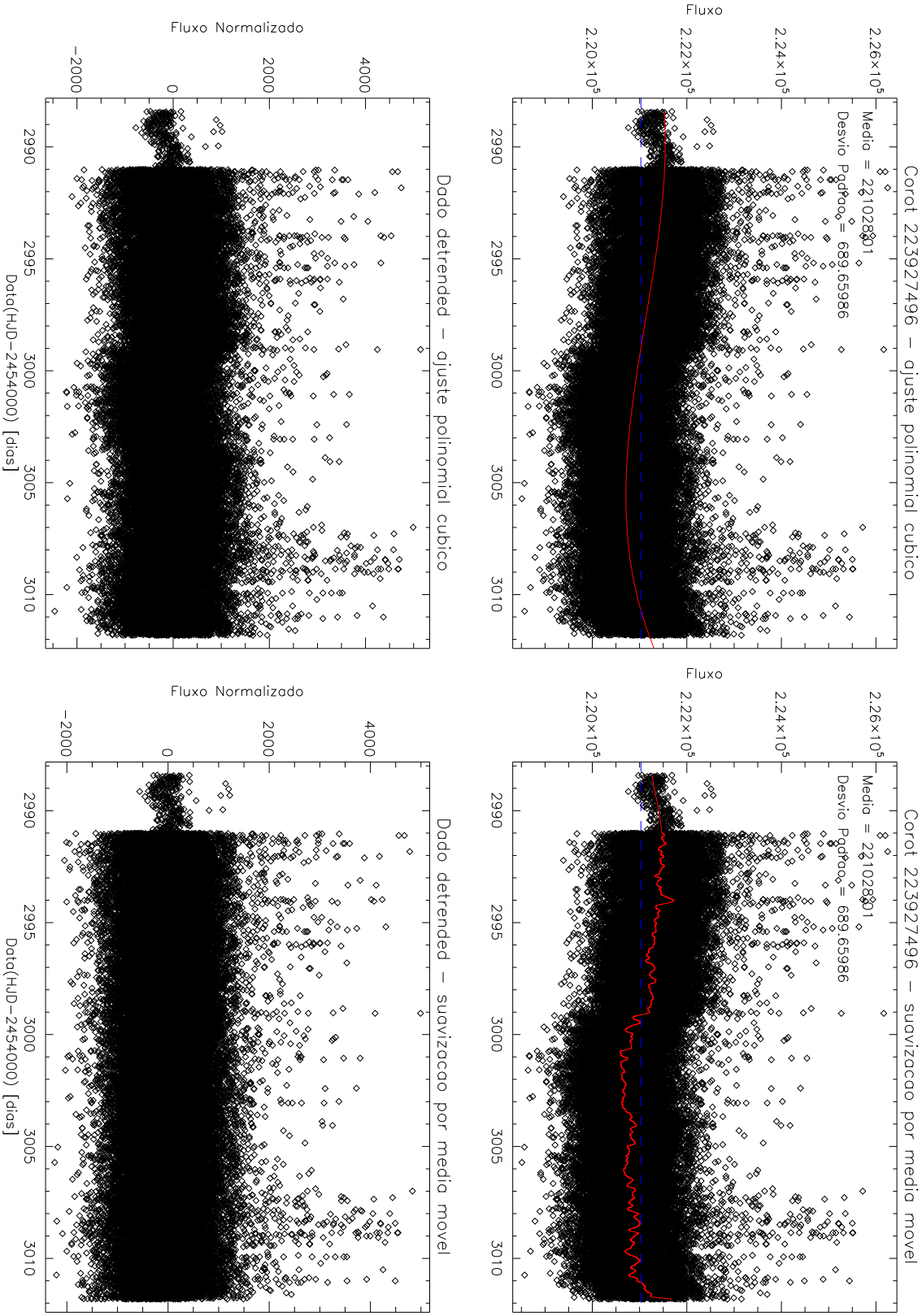


Figura 7.2: Arquivo 0223927496.fits.

A Figura 7.4 apresenta a classificação da série temporal (22) na respectiva classe de séries não-estacionárias, devido à declaração que a mesma apresenta a suposição de não-estacionariedade.

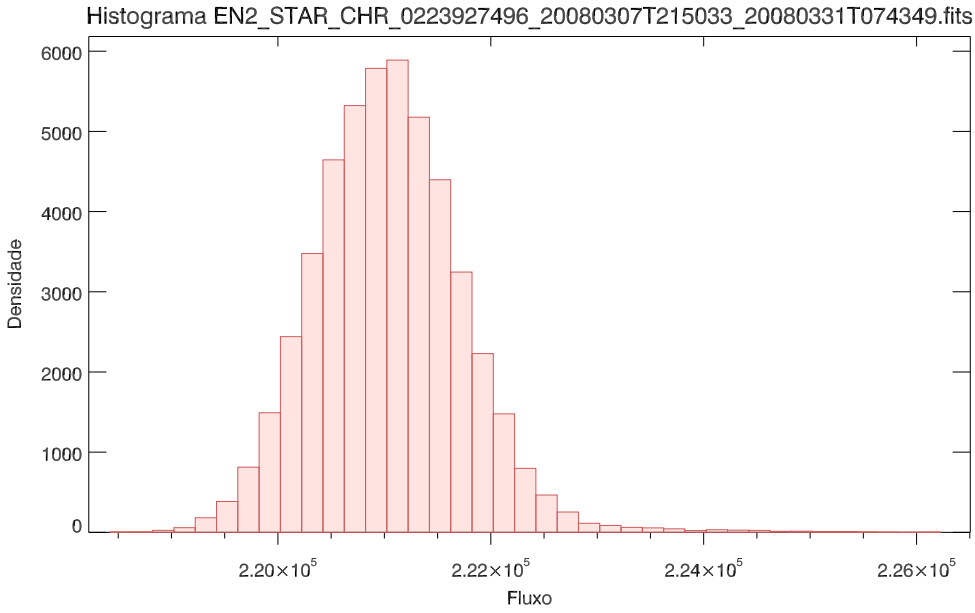


Figura 7.3: Histograma EN2_STAR_CHR_0223927496.fits.

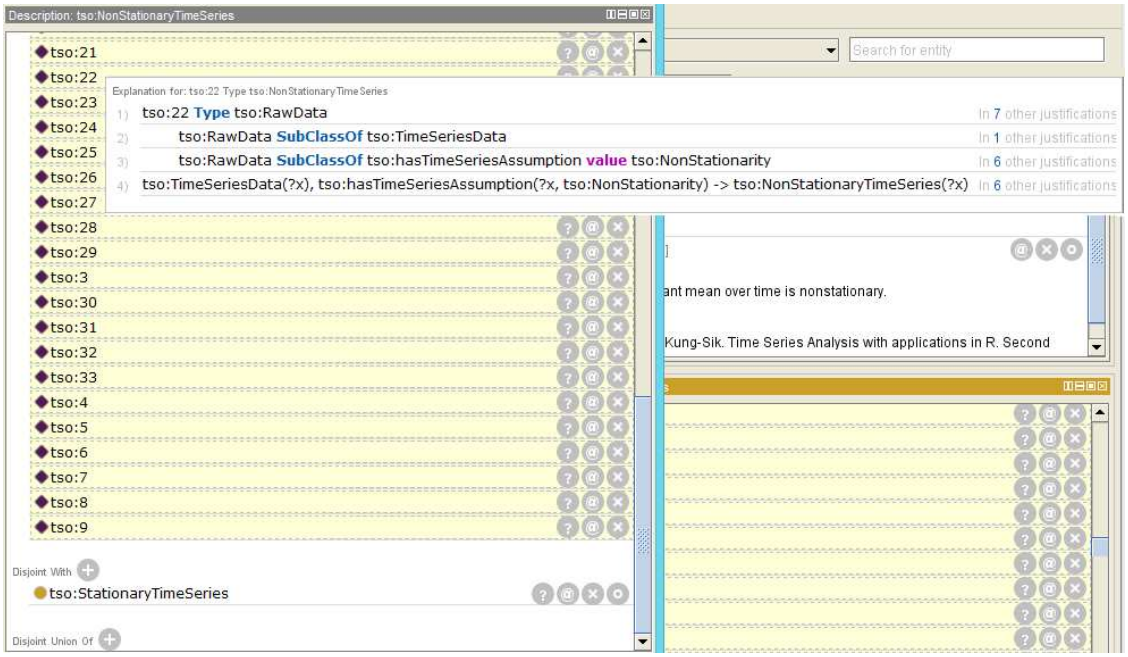


Figura 7.4: Exemplo de inferência nas séries temporais a partir de regras definidas.

A Figura 7.5 apresenta exemplos de inferências obtidas por meio de regras definidas na ontologia TSO e compiladas com o *reasoner* Pellet. No caso da série temporal (22), são declaradas as suposições de: i. Não-estacionariedade, devido à mesma apresentar tendência. ii. Normalidade, conforme o Histograma da Figura 7.3; iii. Não-Linearidade, devido a mesma ser uma série temporal real; e iv. Discreta, sendo uma série obtida em intervalos regulares de tempo, caracterizando uma série temporal homogênea quanto ao tipo de observação; e v. Apresenta comportamento não-monotônico. Assim que declaradas essas suposições, a série é inferida nas respectivas classes da ontologia TSO:

(NonStationaryTimeSeries), (NormalTimeSeries), (NonLinearTimeSeries), (DiscreteTimeSeries) e (HomogeneousTimeSeries) e (NonMonotoneTimeSeries), respectivamente.

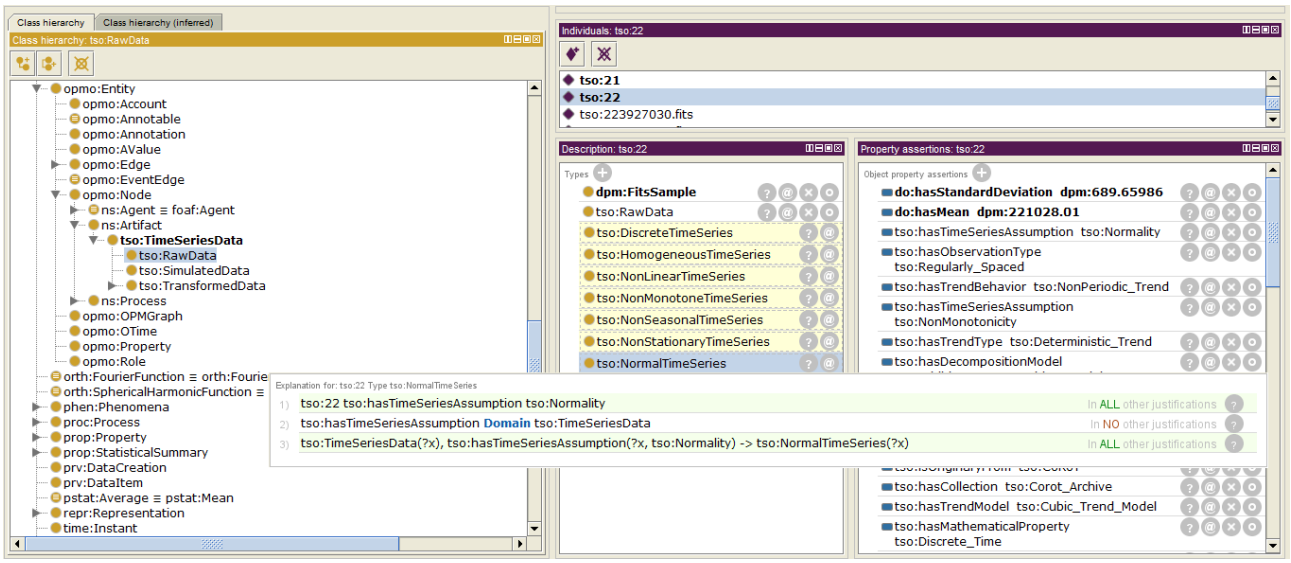


Figura 7.5: Exemplo de inferência da série original (22) nas respectivas classes de séries temporais, conforme regras definidas.

Em um processo de *detrending*, quando uma série é corrigida de tendência, uma nova série é gerada. No caso do método de *detrending* de regressão polinomial, a série *detrended* (220) é inserida na classe (dpm:RegressedDetrendedData), a qual é inferida, devido ao axioma (tso:hasAssumption value tso:Stationarity) na classe (tso:StationaryTimeSeries), caracterizando uma nova série temporal estacionária. Na Figura 7.6 é apresentada a inferência obtida quando as séries temporais foram corrigidas de tendências por meio de regressão.

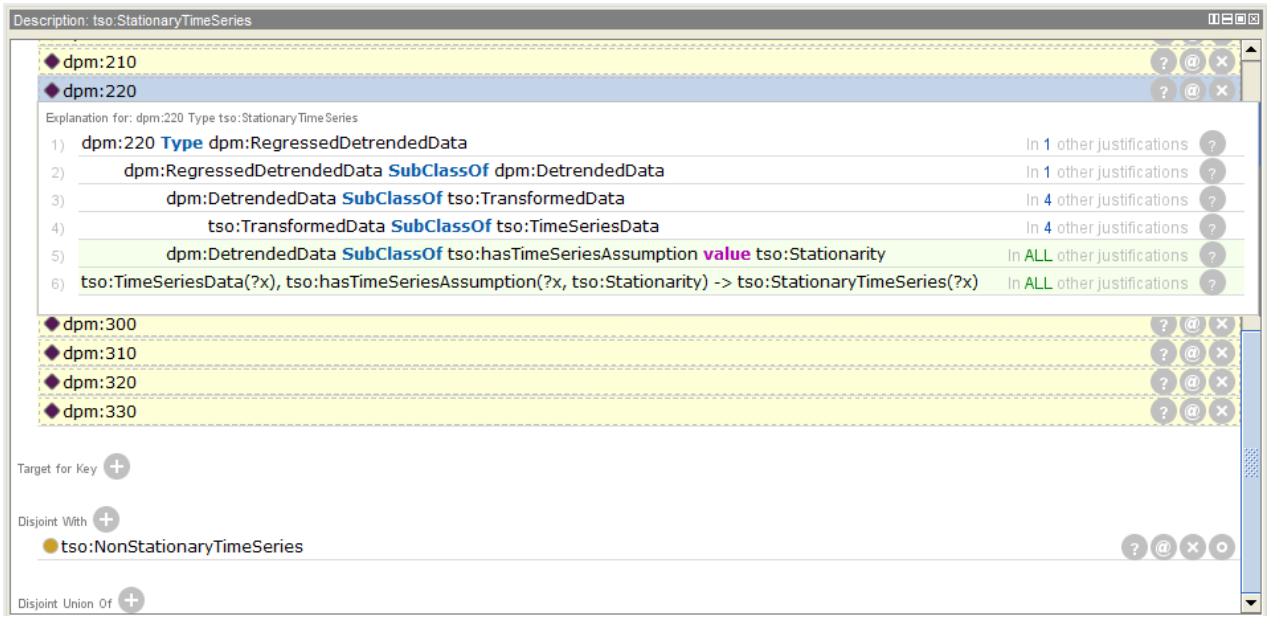


Figura 7.6: Exemplo de inferência de uma série *detrended* em (tso:StationaryTimeSeries).

A Figura 7.7 apresenta as séries temporais corrigidas de tendência por regressão

e o respectivo *Uniform Resource Locator* - URL das séries temporais originais, sendo informado que a tendência é considerada como do tipo determinístico, apresentando modelo de tendência cúbica, período de longo prazo e comportamento não-periódico e não-monotônico, conforme citado em [190].

SPARQL query:

```
SELECT ?detrendeddata ?rawdata ?rawdataurl ?trendmodel ?trendtype ?trendperiod ?trendbehavior
        WHERE { ?wdf opmo:effectWasDerivedFrom ?detrendeddata ;
                    opmo:causeWasDerivedFrom ?rawdata .
                ?rawdata tso:url ?rawdataurl ;
                    tso:hasTrendType ?trendtype ;
                    tso:hasTrendModel ?trendmodel ;
                    tso:hasTrendPeriod ?trendperiod ;
                    tso:hasTrendBehavior ?trendbehavior .} ORDER BY ?rawdata
```

detrendeddata	rawdata	rawdataurl	trendmodel	trendtype	trendperiod	trendbehavior
210	21	"C:\corot\corodata2\SRa01_EN2_V2_1\223927030.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonPeriodic_Trend
210	21	"C:\corot\corodata2\SRa01_EN2_V2_1\223927030.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonMonotonic_Trend
220	22	"C:\corot\corodata2\SRa01_EN2_V2_1\223927496.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonPeriodic_Trend
220	22	"C:\corot\corodata2\SRa01_EN2_V2_1\223927496.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonMonotonic_Trend
230	23	"C:\corot\corodata2\SRa01_EN2_V2_1\223927663.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonPeriodic_Trend
230	23	"C:\corot\corodata2\SRa01_EN2_V2_1\223927663.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonMonotonic_Trend
240	24	"C:\corot\corodata2\SRa01_EN2_V2_1\223927728.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonPeriodic_Trend
240	24	"C:\corot\corodata2\SRa01_EN2_V2_1\223927728.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonMonotonic_Trend
250	25	"C:\corot\corodata2\SRa01_EN2_V2_1\223927797.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonPeriodic_Trend
250	25	"C:\corot\corodata2\SRa01_EN2_V2_1\223927797.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonMonotonic_Trend
260	26	"C:\corot\corodata2\SRa01_EN2_V2_1\223927955.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonMonotonic_Trend
260	26	"C:\corot\corodata2\SRa01_EN2_V2_1\223927955.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonPeriodic_Trend
270	27	"C:\corot\corodata2\SRa01_EN2_V2_1\223929900.fits"^^	<Cubic_Trend_Model	Deterministic_Trend	Long_Term_Trend	NonMonotonic_Trend

Figura 7.7: Consulta sobre séries temporais corrigidas de tendência por regressão, URL das séries originais e informações sobre a tendência.

A Figura 7.8 mostra séries temporais corrigidas de tendência por regressão, o URL das séries originais, o modelo de decomposição considerado e os componentes irregular e tendência.

SPARQL query:

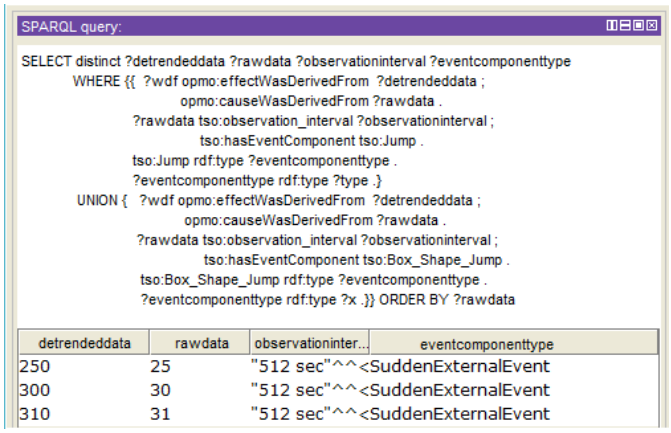
```
SELECT distinct ?detrendeddata ?rawdata ?rawdataurl ?decompositionmodel ?component
        WHERE {  ?wdf opmo:effectWasDerivedFrom  ?detrendeddata ;
                    opmo:causeWasDerivedFrom ?rawdata .
                ?rawdata tso:url ?rawdataurl ;
                    tso:hasDecompositionModel ?decompositionmodel ;
                    tso:hasDecompositionType ?decompositiontype .
                ?decompositiontype tso:hasComponent ?component .} ORDER BY ?rawdata
```

detrendeddata	rawdata	rawdataurl	decompositionmodel	component
210	21	"C:\corot\corodata2\SRa01_EN2_V2_1\223927030.fits"^^	<Additive_Decomposition_Model	Irregular_Component
210	21	"C:\corot\corodata2\SRa01_EN2_V2_1\223927030.fits"^^	<Additive_Decomposition_Model	Trend_Component
220	22	"C:\corot\corodata2\SRa01_EN2_V2_1\223927496.fits"^^	<Additive_Decomposition_Model	Irregular_Component
220	22	"C:\corot\corodata2\SRa01_EN2_V2_1\223927496.fits"^^	<Additive_Decomposition_Model	Trend_Component
230	23	"C:\corot\corodata2\SRa01_EN2_V2_1\223927663.fits"^^	<Additive_Decomposition_Model	Irregular_Component
230	23	"C:\corot\corodata2\SRa01_EN2_V2_1\223927663.fits"^^	<Additive_Decomposition_Model	Trend_Component
240	24	"C:\corot\corodata2\SRa01_EN2_V2_1\223927728.fits"^^	<Additive_Decomposition_Model	Irregular_Component
240	24	"C:\corot\corodata2\SRa01_EN2_V2_1\223927728.fits"^^	<Additive_Decomposition_Model	Trend_Component
250	25	"C:\corot\corodata2\SRa01_EN2_V2_1\223927797.fits"^^	<Additive_Decomposition_Model	Irregular_Component
250	25	"C:\corot\corodata2\SRa01_EN2_V2_1\223927797.fits"^^	<Additive_Decomposition_Model	Trend_Component
260	26	"C:\corot\corodata2\SRa01_EN2_V2_1\223927955.fits"^^	<Additive_Decomposition_Model	Irregular_Component
260	26	"C:\corot\corodata2\SRa01_EN2_V2_1\223927955.fits"^^	<Additive_Decomposition_Model	Trend_Component

Figura 7.8: Consulta sobre séries temporais corrigidas de tendência por regressão, URL das séries originais, modelo de decomposição e componentes.

Mislis et al [190] classificam os *jumps* (saltos aleatórios) que podem ocorrer em séries temporais do tipo FITS em cinco tipos, considerando que os mesmos aparecem em diferentes filtros e em diferentes tempos. Um dos formatos de *jump* que aparece na série temporal de ID (25), armazenada no arquivo físico (0223927797.fits) é o formato caixa. O conhecimento sobre o formato do *jump* (quando aplicável) é relevante para tomada de decisão, pois no caso de um formato de *jump* que diminui a intensidade repentinamente e diminui de forma exponencial é similar a uma estrela *flare* e o algoritmo CDA não se aplica a este tipo de estrela.

Outro caso é quando o *jump* diminui a intensidade repentinamente e aumenta de forma exponencial, sendo similar a um evento de transiente, tornando difícil a detecção do trânsito verdadeiro, entre outras situações. A Figura 7.9 mostra as séries temporais originais e o intervalo de observação, a partir das quais novas séries temporais foram derivadas e *detrended*, as quais contém *jump* em qualquer formato e no formato de caixa.



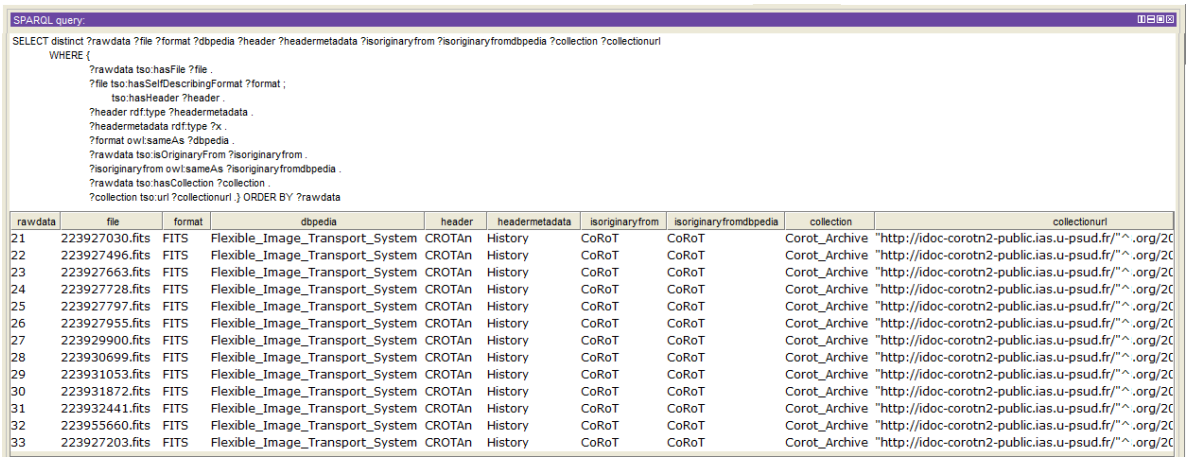
SPARQL query:

```
SELECT distinct ?detrendeddata ?rawdata ?observationinterval ?eventcomponenttype
        WHERE {{  ?wdf opmo:effectWasDerivedFrom  ?detrendeddata ;
                        opmo:causeWasDerivedFrom ?rawdata .
                ?rawdata tso:observation_interval ?observationinterval ;
                        tso:hasEventComponent tso:Jump .
                tso:Jump rdf:type ?eventcomponenttype .
                ?eventcomponenttype rdf:type ?type .}
        UNION {   ?wdf opmo:effectWasDerivedFrom  ?detrendeddata ;
                        opmo:causeWasDerivedFrom ?rawdata .
                ?rawdata tso:observation_interval ?observationinterval ;
                        tso:hasEventComponent tso:Box_Shape_Jump .
                tso:Box_Shape_Jump rdf:type ?eventcomponenttype .
                ?eventcomponenttype rdf:type ?x .}} ORDER BY ?rawdata
```

detrendeddata	rawdata	observationinter...	eventcomponenttype
250	25	"512 sec"^^	<SuddenExternalEvent
300	30	"512 sec"^^	<SuddenExternalEvent
310	31	"512 sec"^^	<SuddenExternalEvent

Figura 7.9: Consulta sobre séries temporais corrigidas a partir de quais séries que apresentam componentes de evento (*jump*), seu tipo e o intervalo de observação das séries temporais originais.

A Figura 7.10 apresenta uma consulta relacionada aos arquivos das séries temporais originais, seu formato e associação do mesmo com a DBpedia. São apresentados os cabeçalhos (*header*) dos arquivos e o tipo do mesmo, nesse caso, apresenta o metadado *History* que é usado para armazenamento de informações de proveniência nos arquivos. A consulta mostra de qual instrumento a série é originária e sua associação com a DBpedia, qual coleção a série pertence e o respectivo URL da coleção.



SPARQL query:

```
SELECT distinct ?rawdata ?file ?format ?dbpedia ?header ?headermetadata ?isoriginaryfrom ?isoriginaryfromdbpedia ?collection ?collectionurl
        WHERE {
                ?rawdata tso:hasFile ?file .
                ?file tso:hasSelfDescribingFormat ?format ;
                      tso:hasHeader ?header .
                ?header rdf:type ?headermetadata .
                ?headermetadata rdf:type ?x .
                ?format owl:sameAs ?dbpedia .
                ?rawdata tso:isOriginaryFrom ?isoriginaryfrom .
                ?isoriginaryfrom owl:sameAs ?isoriginaryfromdbpedia .
                ?rawdata tso:hasCollection ?collection .
                ?collection tso:url ?collectionurl .} ORDER BY ?rawdata
```

rawdata	file	format	dbpedia	header	headermetadata	isoriginaryfrom	isoriginaryfromdbpedia	collection	collectionurl
21	223927030.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
22	223927496.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
23	223927663.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
24	223927728.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
25	223927797.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
26	223927955.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
27	223929900.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
28	223930699.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
29	223931053.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
30	223931872.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
31	223932441.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
32	223955660.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(
33	223927203.fits	FITS	Flexible_Image_Transport_System	CROTAn	History	CoRoT	CoRoT	Corot_Archive	"http://idoc-corotn2-public.ias.u-psud.fr/"^.org/2(

Figura 7.10: Consulta sobre o arquivo, formato, cabeçalho, instrumento científico e coleção das séries originais e associação com a DBpedia.

As Figuras 7.11 a 7.14 são relacionadas ao algoritmo CDA. A Figura 7.11 apresenta o método de *detrending* e sua aplicabilidade no algoritmo CDA, incluindo o domínio do método e sua estatística, associados com a DBpedia, possibilitando gerar mais informações, assim como permitindo interoperabilidade semântica.

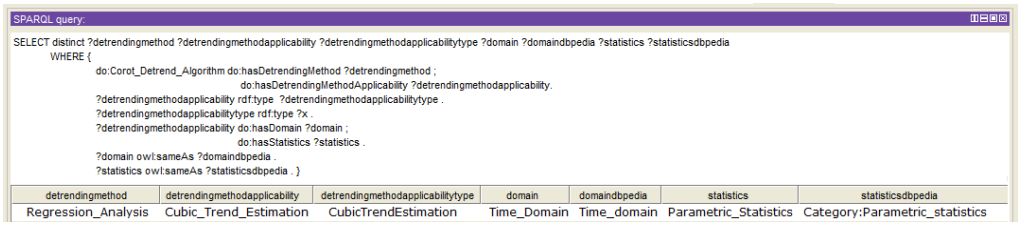


Figura 7.11: Consulta sobre o método de *detrending* e sua aplicabilidade em CDA, domínio do método, estatística e associação com a DBpedia.

A Figura 7.12 apresenta o acrônimo e a versão do algoritmo CDA, o software relacionado e em qual linguagem de programação o mesmo é implementado, apresentando a associação com a DBpedia. A Figura 7.13 apresenta os relacionamentos do algoritmo CDA e suas instâncias.

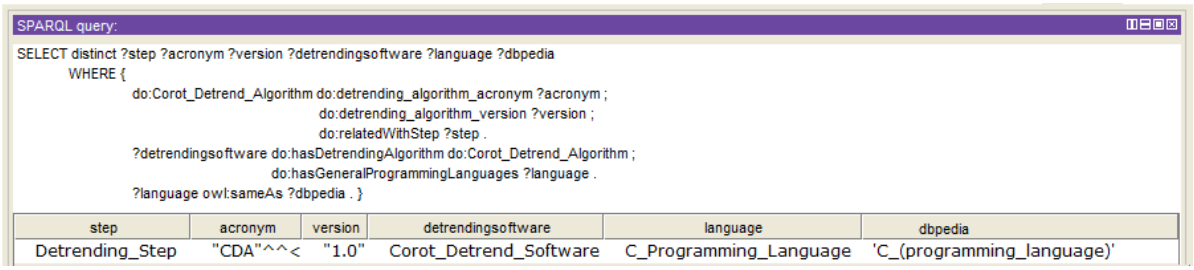


Figura 7.12: Consulta sobre o acrônimo e versão do algoritmo CDA, software relacionado e linguagem de programação, incluindo a associação com a DBpedia.

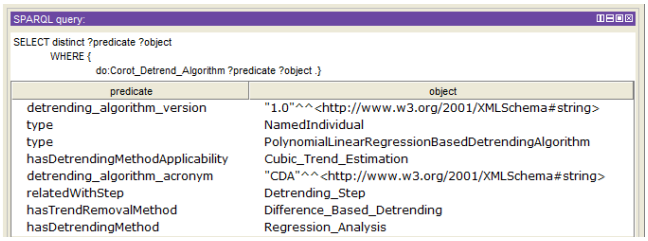


Figura 7.13: Consulta sobre os relacionamentos do algoritmo CDA e suas instâncias.

Como a tendência é um componente de baixa frequência, após sua estimação de forma paramétrica, é necessário fazer uso de uma operação binária para sua extração. Nesse caso, devido ao modelo de decomposição das séries ser considerado aditivo, conforme a Figura 7.8 a mesma é subtraída a partir das séries, como é o caso do algoritmo CDA (Figura 7.14).

A Figura 7.15 apresenta as restrições de domínio e intervalo a partir da instanciação WCB e a associação do agente de *detrending* com o respectivo software e o algoritmo CDA, sendo apresentado o tipo do algoritmo, o método utilizado, sua aplicabilidade e o tipo de análise relacionada.

A Figura 7.16 apresenta as restrições de domínio e intervalo de séries temporais geradas e corrigidas de tendência por sua estimação, apresentando o respectivo processo e o tipo de dado gerado e corrigido por regressão.

A Figura 7.17 apresenta um relacionamento de causa e efeito da classe Used, apresentando as séries temporais que foram usadas por um processo de *detrending* e os respectivos

SPARQL query:

```
prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>
prefix owl:<http://www.w3.org/2002/07/owl#>
prefix dc:<http://purl.org/dc/elements/1.1/>
prefix do:<http://www.semanticweb.org/ontologies/2013/7/DetrendOntology.owl#>
prefix xsd:<http://www.w3.org/2001/XMLSchema#>
prefix dbpedia:<http://dbpedia.org/>

SELECT distinct ?detrendingmethod ?detrendingmethodapplicability ?trendremoval ?trendremovalclass ?operation
WHERE{ do:Corot_Detrend_Algorithm do:hasDetrendingMethod ?detrendingmethod ;
                                  do:hasDetrendingMethodApplicability ?detrendingmethodapplicability ;
                                  do:hasTrendRemovalMethod ?trendremoval .
                                  ?trendremoval rdf:type ?trendremovalclass .
                                  ?trendremovalclass rdf:type ?x .
                                  ?instance rdf:type ?trendremovalclass ;
                                  do:hasBinaryOperation ?operation .}
```

detrendingmethod	detrendingmethodapplicability	trendremoval	trendremovalclass	operation
Regression_Analysis	Cubic_Trend_Estimation	Difference_Based_Detrending	DifferenceBasedDetrending	Subtraction

Figura 7.14: Consulta sobre o método de remoção da tendência do algoritmo CDA.

SPARQL query:

```
SELECT distinct ?detrendingprocess ?detrendingagent ?detrendingsoftware ?detrendingmethod ?detrendingmethodapplicability ?detrendingalgorithmtype ?analysis
        WHERE {
            ?wcb opmo:effectWasControlledBy ?detrendingprocess ;
                 opmo:causeWasControlledBy ?detrendingagent .
            ?detrendingagent dpm:hasDetrendingSoftware ?detrendingsoftware .
            ?detrendingsoftware do:hasDetrendingAlgorithm do:Corot_Detrend_Algorithm .
            do:Corot_Detrend_Algorithm rdf:type ?detrendingalgorithmtype ;
                                       do:hasDetrendingMethod ?detrendingmethod ;
                                       do:hasDetrendingMethodApplicability ?detrendingmethodapplicability .
            ?detrendingmethodapplicability do:hasAnalysis ?analysis .
            ?detrendingalgorithmtype rdf:type ?x .}
```

detrendingprocess	detrendingagent	detrendingsoftware	detrendingmethod	detrendingmethodapplicability	detrendingalgorithmtype	analysis
Regression_Based_Detrending_Proc	Regression_Based_Detrending_Software_Ag	Corot_Detrend_Software	Regression_Analysis	Cubic_Trend_Estimation	PolynomialLinearReg	Cubic_Regression

Figura 7.15: Consulta sobre o domínio e intervalo de WCB, associação do agente de *detrending* com o software e o algoritmo CDA e seus relacionamentos.

SPARQL query:

```
SELECT distinct ?timeseriesdata ?detrendingprocess ?timeseriesdatatype
   WHERE {   ?wgdb dpm:effectWasGeneratedDetrendedBy  ?timeseriesdata ;
                   dpm:causeWasGeneratedDetrendedBy ?detrendingprocess .
             ?timeseriesdata rdf:type ?timeseriesdatatype .
             ?timeseriesdatatype rdf:type ?x .} ORDER By ?timeseriesdata
```

timeseriesdata	detrendingprocess	timeseriesdatatype
210	Regression_Based_Detrending_Process	RegressedDetrendedData
220	Regression_Based_Detrending_Process	RegressedDetrendedData
230	Regression_Based_Detrending_Process	RegressedDetrendedData
240	Regression_Based_Detrending_Process	RegressedDetrendedData
250	Regression_Based_Detrending_Process	RegressedDetrendedData
260	Regression_Based_Detrending_Process	RegressedDetrendedData
270	Regression_Based_Detrending_Process	RegressedDetrendedData
280	Regression_Based_Detrending_Process	RegressedDetrendedData
290	Regression_Based_Detrending_Process	RegressedDetrendedData
300	Regression_Based_Detrending_Process	RegressedDetrendedData
310	Regression_Based_Detrending_Process	RegressedDetrendedData
320	Regression_Based_Detrending_Process	RegressedDetrendedData
330	Regression_Based_Detrending_Process	RegressedDetrendedData

Figura 7.16: Consulta sobre o domínio e intervalo de WGDB das séries geradas e corrigidas por regressão e seu tipo.

tipos dos mesmos.

A consulta da Figura 7.18 mostra as séries derivadas a partir das séries originais, com seu respectivo URL, propriedade matemática associada, intervalo de observação, domínio do conhecimento relacionado e sua associação com a DBpedia, apresentando as suposições consideradas das séries temporais originais.

A consulta da Figura 7.19 apresenta as séries temporais derivadas e corrigidas de tendência a partir das séries originais, mostrando o URL, a média e o desvio padrão das mesmas, correspondendo às medidas presentes nas séries temporais originais CoRoT.

SPARQL query:

```
SELECT distinct ?detrendingprocess ?detrendingprocesstype ?timeseriesdata ?timeseriesdatatype
WHERE { ?used opmo:effectUsed ?detrendingprocess ;
              opmo:causeUsed ?timeseriesdata .
              ?timeseriesdata rdf:type ?timeseriesdatatype .
              ?timeseriesdatatype rdf:type ?x .
              ?detrendingprocess rdf:type ?detrendingprocesstype .
               ?detrendingprocesstype rdf:type ?y .} ORDER By ?timeseriesdata
```

detrendingprocess	detrendingprocesstype	timeseriesdata	timeseriesdatatype
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	23	RawData
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	24	NonLinearTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	24	DiscreteTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	24	NonSeasonalTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	24	FitsSample
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	24	NormalTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	24	NonMonotoneTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	24	HomogeneousTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	24	InTheMeanNonStationaryTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	24	RawData
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	25	DiscreteTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	25	NonMonotoneTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	25	HomogeneousTimeSeries
Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess	25	NormalTimeSeries

Figura 7.17: Consulta sobre o domínio e intervalo de *Used* entre o processo de *detrending*, as séries e seus tipos.

SPARQL query:

```
SELECT distinct ?detrendeddata ?rawdata ?rawdataurl ?assumption ?observationinterval ?mathematicalproperty ?knowledgedomain ?dbpedia
        WHERE { ?wdf opmo:causeWasDerivedFrom ?rawdata ;
                     opmo:effectWasDerivedFrom ?detrendeddata .
                ?rawdata tso:url ?rawdataurl ;
                         tso:hasTimeSeriesAssumption ?assumption ;
                         tso:observation_interval ?observationinterval ;
                         tso:hasMathematicalProperty ?mathematicalproperty ;
                         tso:hasKnowledgeDomain ?knowledgedomain .
                ?knowledgedomain owl:sameAs ?dbpedia . } ORDER BY ?rawdata
```

detrendeddata	rawdata	rawdataurl	assumption	observationinterval	mathematicalproperty	knowledgedomain	dbpedia
210	21	"C:\corot\corodata2\SRa01_EN2_V2_1\223927030.fits"	Normality	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
210	21	"C:\corot\corodata2\SRa01_EN2_V2_1\223927030.fits"	NonLinearity	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
210	21	"C:\corot\corodata2\SRa01_EN2_V2_1\223927030.fits"	NonSeasonality	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
210	21	"C:\corot\corodata2\SRa01_EN2_V2_1\223927030.fits"	NonMonotonicity	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
220	22	"C:\corot\corodata2\SRa01_EN2_V2_1\223927496.fits"	NonLinearity	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
220	22	"C:\corot\corodata2\SRa01_EN2_V2_1\223927496.fits"	NonMonotonicity	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
220	22	"C:\corot\corodata2\SRa01_EN2_V2_1\223927496.fits"	Normality	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
220	22	"C:\corot\corodata2\SRa01_EN2_V2_1\223927496.fits"	NonSeasonality	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
230	23	"C:\corot\corodata2\SRa01_EN2_V2_1\223927663.fits"	NonMonotonicity	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
230	23	"C:\corot\corodata2\SRa01_EN2_V2_1\223927663.fits"	Normality	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
230	23	"C:\corot\corodata2\SRa01_EN2_V2_1\223927663.fits"	NonLinearity	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
230	23	"C:\corot\corodata2\SRa01_EN2_V2_1\223927663.fits"	NonSeasonality	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science
240	24	"C:\corot\corodata2\SRa01_EN2_V2_1\223927728.fits"	NonMonotonicity	"512 sec"^^	Discrete_Time	Planetary_Science	Category:Planetary_science

Figura 7.18: Consulta sobre séries originais que foram derivadas e corrigidas de tendências por regressão, URL, intervalo de observação, propriedade matemática, domínio de conhecimento e associação com a DBpedia e suposições consideradas.

A Figura 7.20 descreve as séries temporais corrigidas de tendência por regressão e os respectivos tipos das séries temporais originais, classificadas por meio de regras definidas conforme declarações do pesquisador na base de conhecimento.

A Figura 7.21 apresenta os relacionamentods das classes Used, WTB e WCB e seus relacionamentos e o respectivo software e algoritmo de *detrending* CDA.

A Figura 7.22 apresenta as relações WTB, WCD, Used, WDF e WGDB e o respectivo software e a linguagem de programação, o método usado pelo algoritmo e sua aplicabilidade, assim como o intervalo de observação da série temporal de ID (25).

A Figura 7.23 mostra uma consulta envolvendo os relacionamentos WTB, WCB, Used e WDF, o software e o algoritmo relacionados, assim como apresenta informações sobre proveniência das séries temporais.

SPARQL query:

```
SELECT distinct ?detrendeddata ?rawdata ?rawdataurl ?rawdatamean ?rawdatastandarddeviation
        WHERE { ?wdf opmo:causeWasDerivedFrom ?rawdata ;
                     opmo:effectWasDerivedFrom ?detrendeddata .
                ?rawdata tso:url ?rawdataurl ;
                         do:hasMean ?rawdatamean ;
                         do:hasStandardDeviation ?rawdatastandarddeviation . } ORDER BY ?rawdata
```

detrendeddata	rawdata	rawdataurl	rawdatamean	rawdatastandarddeviation
210	21	"C:\corot\corodata2\SRa01_EN2_V2_1\223927030.fits"^	341845.05	804.6783.4
220	22	"C:\corot\corodata2\SRa01_EN2_V2_1\223927496.fits"^	221028.01	689.65986
230	23	"C:\corot\corodata2\SRa01_EN2_V2_1\223927663.fits"^	78492.894	189.09027
240	24	"C:\corot\corodata2\SRa01_EN2_V2_1\223927728.fits"^	83809.901	242.39876
250	25	"C:\corot\corodata2\SRa01_EN2_V2_1\223927797.fits"^	126770.68	239.13654
260	26	"C:\corot\corodata2\SRa01_EN2_V2_1\223927955.fits"^	296813.52	540.18473
270	27	"C:\corot\corodata2\SRa01_EN2_V2_1\223929900.fits"^	146596.05	592.02348
280	28	"C:\corot\corodata2\SRa01_EN2_V2_1\223930699.fits"^	690862.64	6447.9461
290	29	"C:\corot\corodata2\SRa01_EN2_V2_1\223931053.fits"^	133138.89	394.59918
300	30	"C:\corot\corodata2\SRa01_EN2_V2_1\223931872.fits"^	179222.19	966.07799
310	31	"C:\corot\corodata2\SRa01_EN2_V2_1\223932441.fits"^	190836.09	1380.5077
320	32	"C:\corot\corodata2\SRa01_EN2_V2_1\223955660.fits"^	120928.04	346.44255
330	33	"C:\corot\corodata2\SRa01_EN2_V2_1\223927203.fits"^	85352.794	307.22036

Figura 7.19: Consulta sobre a média e desvio padrão das séries temporais que foram corrigidas por regressão.

SPARQL query:

```
SELECT distinct ?detrendeddata ?detrendeddatatype ?rawdata ?rawdatatype
        WHERE { ?wdf opmo:causeWasDerivedFrom ?rawdata ;
                     opmo:effectWasDerivedFrom ?detrendeddata .
                ?detrendeddata rdf:type ?detrendeddatatype .
                ?detrendeddatatype rdf:type ?type .
                ?rawdata rdf:type ?rawdatatype .
                ?rawdatatype rdf:type ?x .} ORDER BY ?rawdata
```

detrendeddata	detrendeddatatype	rawdata	rawdatatype
250	RegressedDetrendedData	25	DiscreteTimeSeries
250	RegressedDetrendedData	25	NonMonotoneTimeSeries
250	RegressedDetrendedData	25	HomogeneousTimeSeries
250	RegressedDetrendedData	25	NormalTimeSeries
250	RegressedDetrendedData	25	NonLinearTimeSeries
250	RegressedDetrendedData	25	InTheMeanNonStationaryTimeSeries
250	RegressedDetrendedData	25	FitsSample
250	RegressedDetrendedData	25	NonSeasonalTimeSeries
250	RegressedDetrendedData	25	RawData
260	RegressedDetrendedData	26	FitsSample
260	RegressedDetrendedData	26	RawData
270	RegressedDetrendedData	27	FitsSample
270	RegressedDetrendedData	27	RawData

Figura 7.20: Consulta sobre séries temporais corrigidas de tendência por regressão e tipos das séries originais, classificadas por regras definidas.

SPARQL query:

```
SELECT distinct ?detrendingprocess ?triggeredprocess ?detrendingprocess ?detrendingagent ?detrendeddata ?rawdata ?detrendingsoftware ?detrendingalgorithm
        WHERE {
                ?wastriggereddetrendedby opmo:effectWasTriggeredBy ?detrendingprocess ;
                                         opmo:causeWasTriggeredBy ?triggeredprocess .
                ?wascontrolleddetrendedby opmo:effectWasControlledBy ?detrendingprocess ;
                                          opmo:causeWasControlledBy ?detrendingagent .
                ?used opmo:effectUsed ?detrendingprocess ;
                      opmo:causeUsed?rawdata .
                ?wdf opmo:effectWasDerivedFrom ?detrendeddata ;
                     opmo:causeWasDerivedFrom ?rawdata .
                ?detrendingagent dpm:hasDetrendingSoftware ?detrendingsoftware .
                ?detrendingsoftware do:hasDetrendingAlgorithm ?detrendingalgorithm .} ORDER BY ?rawdata
```

detrendingprocess	triggeredprocess	detrendingagent	detrendeddata	rawdata	detrendingsoftware	detrendingalgorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	210	21	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	220	22	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	230	23	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	240	24	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	250	25	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	260	26	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	270	27	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	280	28	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	290	29	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	300	30	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	310	31	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	320	32	Corot_Detrend_Software	Corot_Detrend_Algorithm
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent	330	33	Corot_Detrend_Software	Corot_Detrend_Algorithm

Figura 7.21: Consulta sobre as classes Used, WTB e WCB, seus relacionamentos e o software e algoritmo de *detrending*.

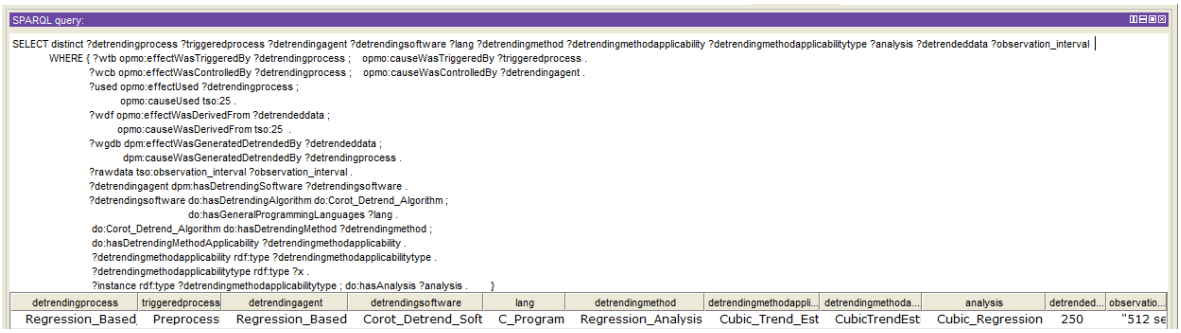


Figura 7.22: Consulta sobre as classes WTB, WCD, Used, WDF e WGDB e o software, a linguagem, o método usado pelo algoritmo e sua aplicabilidade e o intervalo de observação de uma dada série temporal.

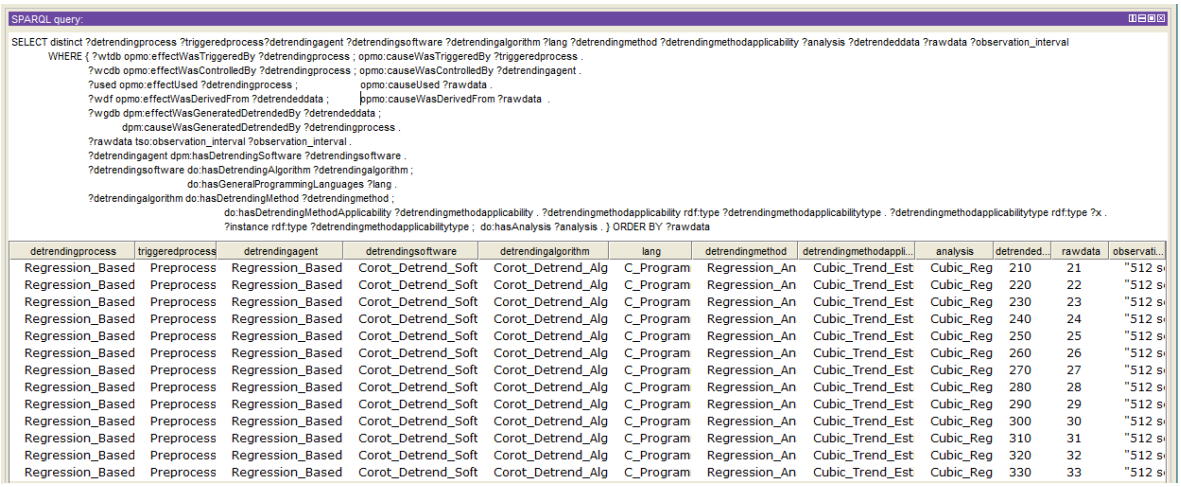


Figura 7.23: Consulta sobre as instâncias das classes WTB, WCB, Used, WDF, software e algoritmo e a proveniência das séries temporais.

A Figura 7.24 mostra uma consulta sobre os relacionamentos WTB, WCB, software, linguagem de programação e o algoritmo relacionados, incluindo o método, sua aplicabilidade e seu tipo e qual a análise relacionada. Apresenta também os dados *detrended* e fornece informações de proveniência quanto às séries temporais originais, tais como o instrumento científico relacionado, as suposições consideradas e o tipo da tendência.

As Figuras 7.25, 7.26 e 7.27 apresentam consultas relacionadas ao grafo de proveniência CDA_Graph. A Figura 7.28 mostra os relacionamentos da série temporal (24) e a respectiva série temporal derivada e corrigida de tendência (240) e seus relacionamentos.

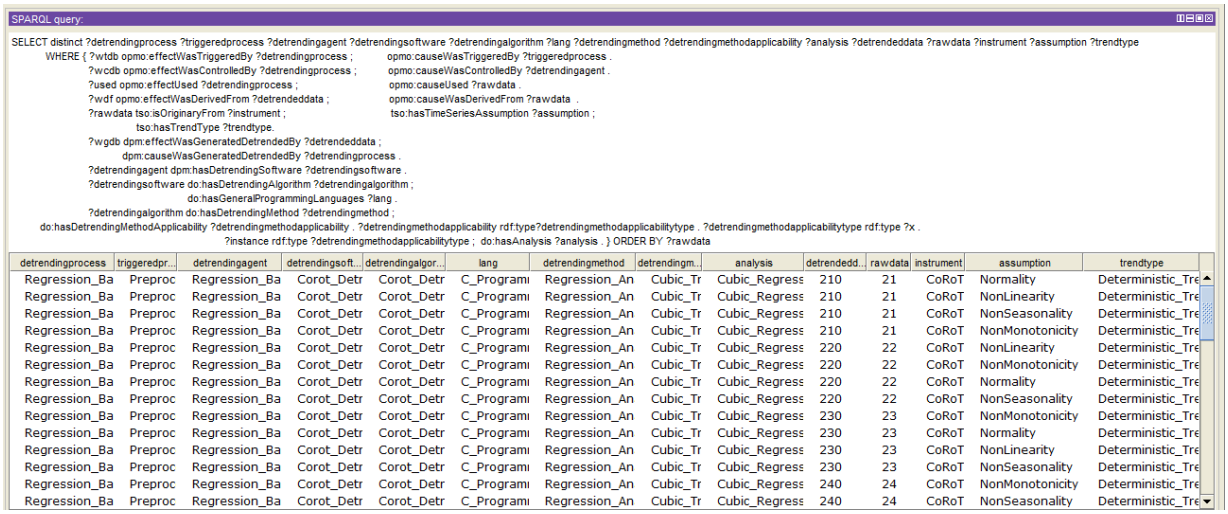


Figura 7.24: Consulta sobre o processo de *detrending* e o processo que o disparou, agente, software e linguagem de programação, algoritmo, método, aplicabilidade e seu tipo, análise, dado *detrended* e informações de proveniência das séries temporais originais.

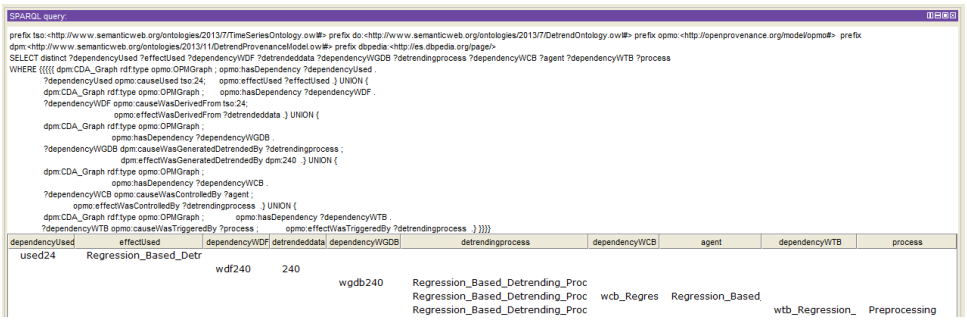


Figura 7.25: Consulta sobre um grafo de proveniência de uma série, processo, agente e dependências.

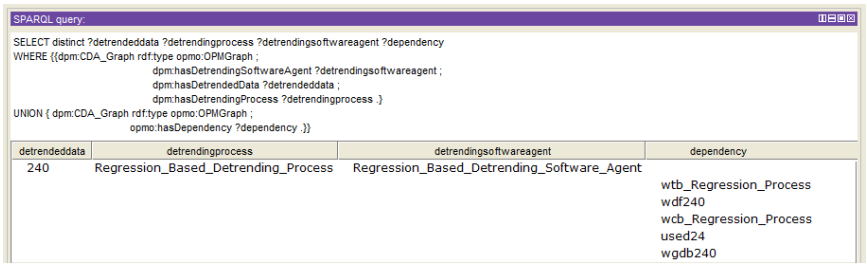


Figura 7.26: Grafo de proveniência de uma série, relações de causa e efeito de dependências.

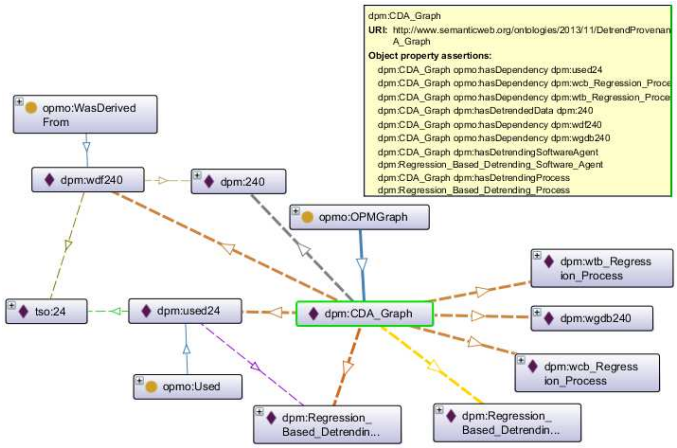


Figura 7.27: Grafo de proveniência de uma dada série e suas dependências.

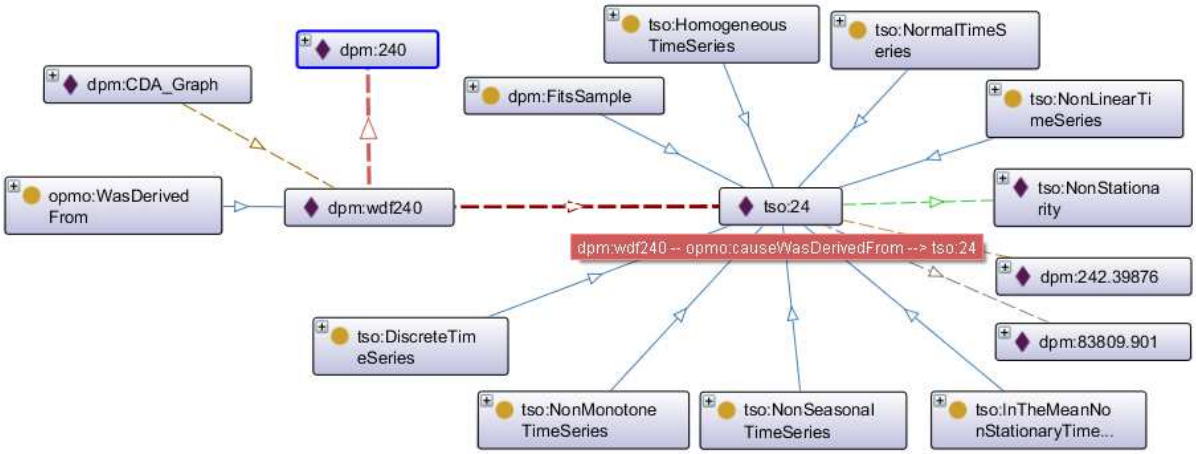


Figura 7.28: Relacionamentos da série (24) e a série temporal derivada e corrigida de tendência (240) e seus relacionamentos.

Na sequência, é apresentado um segundo Caso de Uso, aplicado para as mesmas séries temporais de arquivos FITS, utilizando um outro algoritmo e método para *detrending*, alcançando, segundo pesquisadores da área, resultados significativos em *detrending* para análises posteriores nos dados.

- Caso de Uso 2. *Detrending* usando filtro de média móvel robusta

Este segundo caso de uso está relacionado com a aplicabilidade de um algoritmo desenvolvido a partir do algoritmo CDA [190] do estudo de caso anterior, porém, utiliza o método de média móvel, permitindo melhor identificação da tendência. Uma vez identificada, a mesma é extraída a partir das séries temporais, assim como ocorre com a estimação da tendência de forma paramétrica usando regressão. Nesse caso, o método de estimação é não-paramétrico, onde não ocorre um ajuste de modelo aos dados e a tendência é considerada como uma função suave, a partir dos dados originais que apresentavam flutuações irregulares juntamente com a tendência.

O algoritmo de *detrending* relacionado é *CoRoT Detrend Algorithm Modified* (CDA-M) [78] que usa o método de média móvel robusta para *detrending*. Para sua aplicabilidade, são consideradas as mesmas suposições do algoritmo CDA. O algoritmo CDA-M é aplicado para a mesma amostra (SRa01).

A média móvel robusta utilizada nesse caso é um método de filtro linear, de resposta de impulso finito, o qual bloqueia componentes de alta frequência, como o ruído e permite a passagem de componentes de baixa frequência, como a tendência, caracterizado como um filtro de suavização.

7.2.2 Consultas Relacionadas ao Caso de Uso 2

Nesse caso de uso, as séries temporais originais são consideradas as mesmas do estudo de caso anterior, assim como suas suposições. O tipo da tendência foi alterado para tendência suave, e o comportamento e o período são os mesmos. Nesse caso, não existe um modelo global para a tendência, dado o uso de um método de estimação não-paramétrico. Por utilizar um método robusto, sendo resistente a *outliers* nos dados, o desvio-padrão considerado é 0.5.

A seguir são apresentadas algumas consultas relacionadas a esse estudo de caso, seguidas por consultas envolvendo os dois algoritmos de *detrending*. A Figura 7.29 apresenta qual o arquivo físico das séries temporais e o tipo de observação, assim como as características da tendência tais como tipo, período e comportamento. Nesse caso não é ajustado um modelo para a tendência, visto que a mesma é considerada como uma função suave. As séries corrigidas e seu tipo também são apresentados.

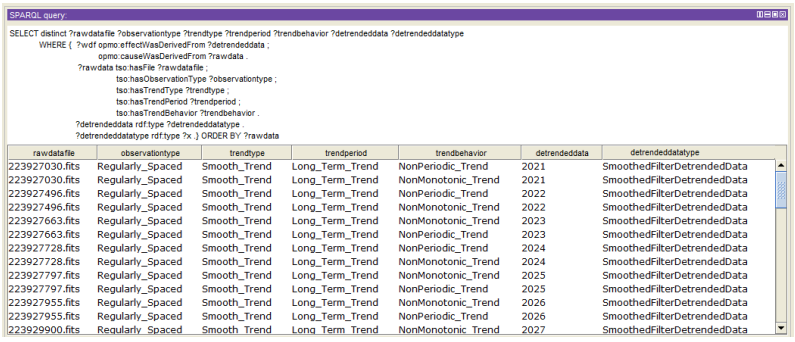


Figura 7.29: Consulta sobre o arquivo e tipo de observação, tipo, período e comportamento da tendência e as séries corrigidas de tendência e seu tipo.

A Figura 7.30 mostra a aplicabilidade do método de suavização baseado em filtro no respectivo algoritmo CDA-M e seus parâmetros.

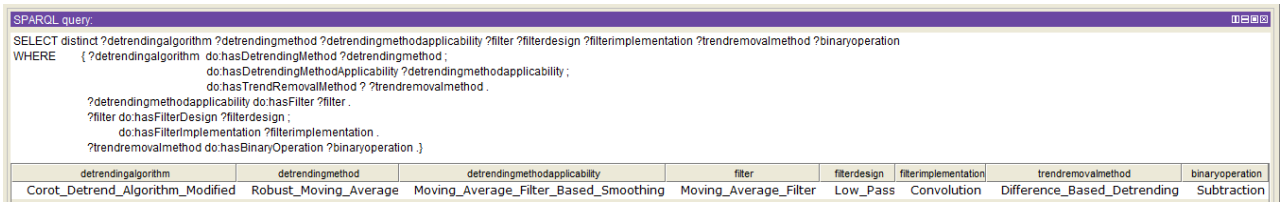


Figura 7.30: Consulta sobre a aplicabilidade dos métodos de suavização baseados em filtros e seus parâmetros.

A Figura 7.31 mostra o processo de suavização que usou as séries temporais e seus tipos e a Figura 7.32 mostra quais séries temporais foram suavizadas e *detrended* por um processo de *detrending* baseado em suavização e seus tipos.

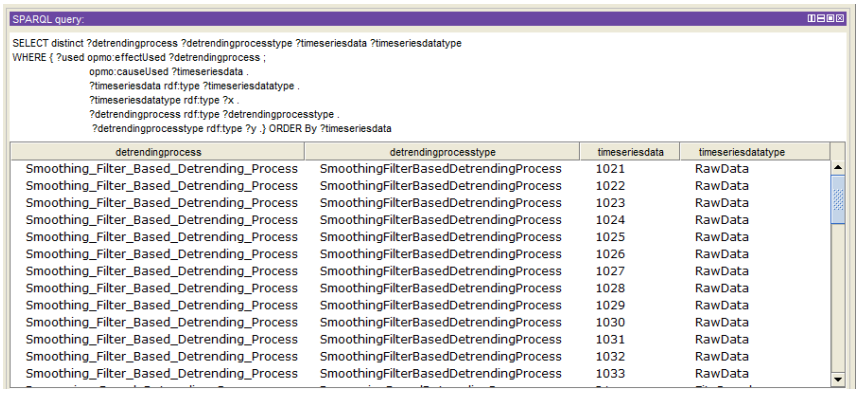


Figura 7.31: Consulta sobre a instância da classe Used e relacionamentos de domínio e intervalo com artefatos e processos e seus tipos.

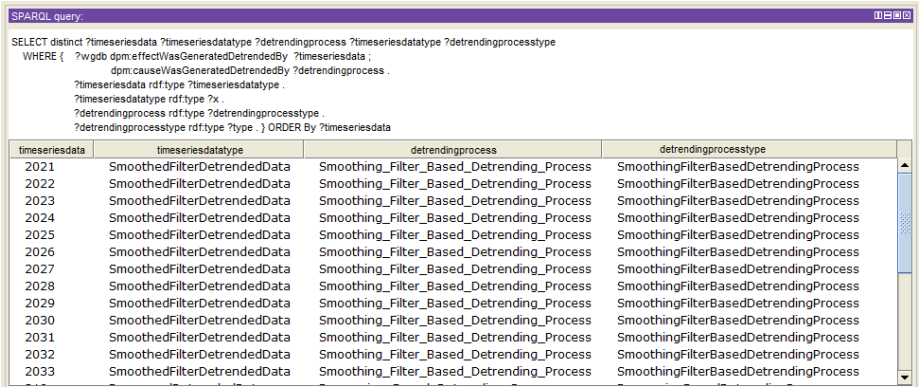


Figura 7.32: Consulta sobre quais séries temporais suavizadas foram geradas e *detrended* a partir de um processo de *detrending* basedo em filtro de suavização e seus tipos.

A Figura 7.33 apresenta o agente que disparou um processo de *detrending*, qual o software e o tipo do algoritmo relacionados. A Figura 7.34 mostra os processos de *detrending* que foram disparados por outros processos e estes foram controlados por quais agentes, qual o software, algoritmo e seu tipo e qual o método utilizado pelo algoritmo CDA-M. A Figura 7.35 mostra instâncias da classe WCDB, o software, o método usado pelo algoritmo CDA-M, sua aplicabilidade e o filtro relacionado. A Figura 7.36 mostra os relacionamentos do algoritmo CDA-M e suas instâncias.

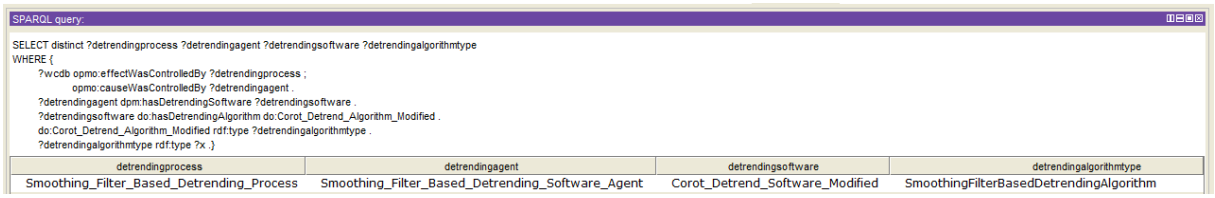


Figura 7.33: Consulta sobre o domínio e intervalo de WCB, agente de *detrending* e o software e o tipo do algoritmo.

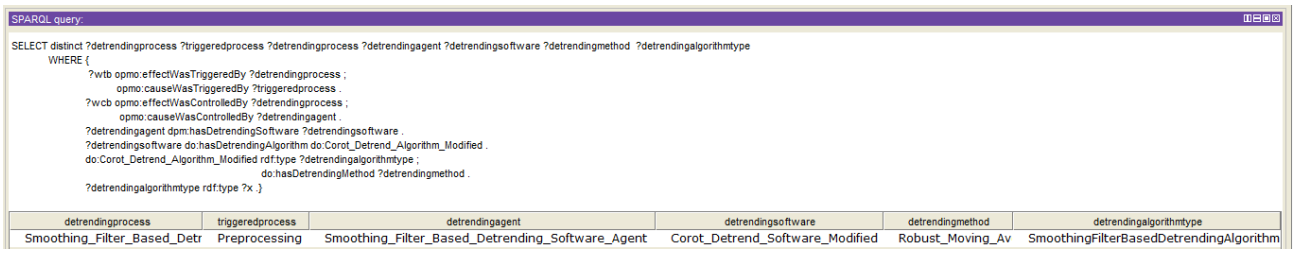


Figura 7.34: Consulta sobre quais processos de *detrending* foram disparados por outros processos e agente que controlou, software e o tipo do algoritmo e o método usado pelo algoritmo CDA-M.

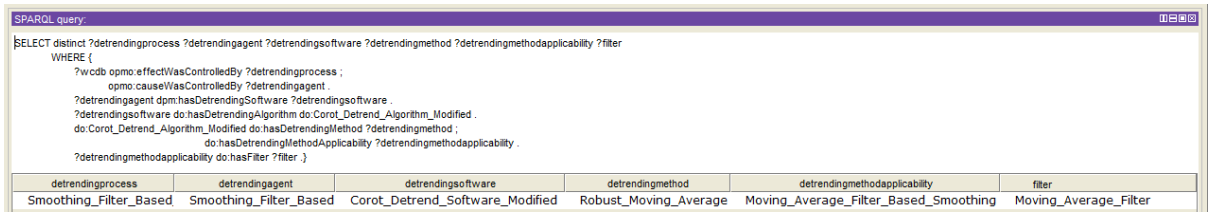


Figura 7.35: Consulta sobre a instância WCDB, o software, o método usado pelo algoritmo CDA-M, sua aplicabilidade e o filtro relacionado.

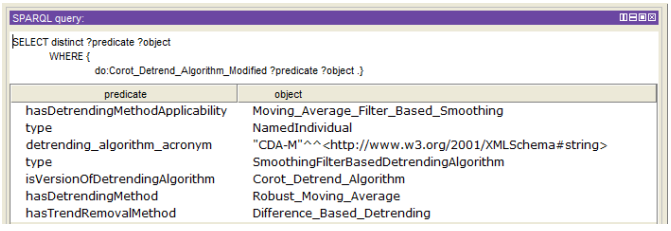


Figura 7.36: Relacionamentos do algoritmo CDA-M e suas instâncias.

As Figuras 7.37 e 7.38 apresentam consultas sobre o grafo de proveniência. A 7.39 apresenta um exemplo de um grafo de proveniência CDA_M e seus relacionamentos e a Figura 7.40 mostra uma consulta sobre os grafos CDA e CDA-M e os respectivos nodos e suas dependências.

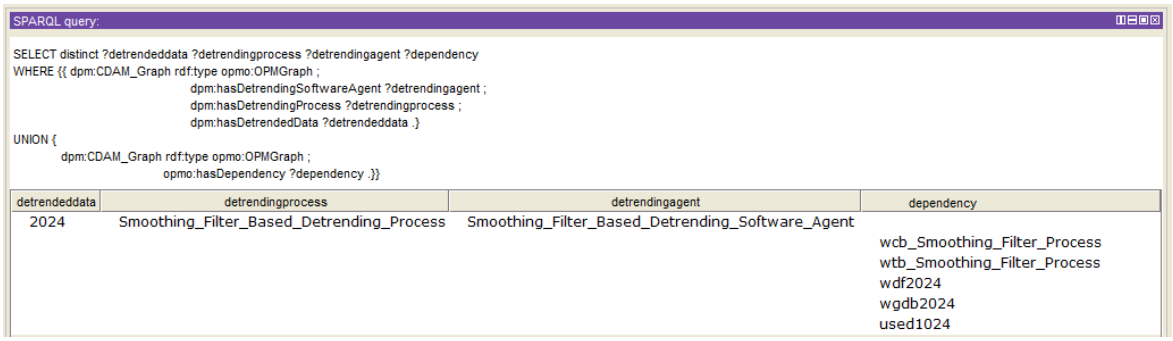


Figura 7.37: Grafo de proveniência de uma série, processo, agente e dependências.

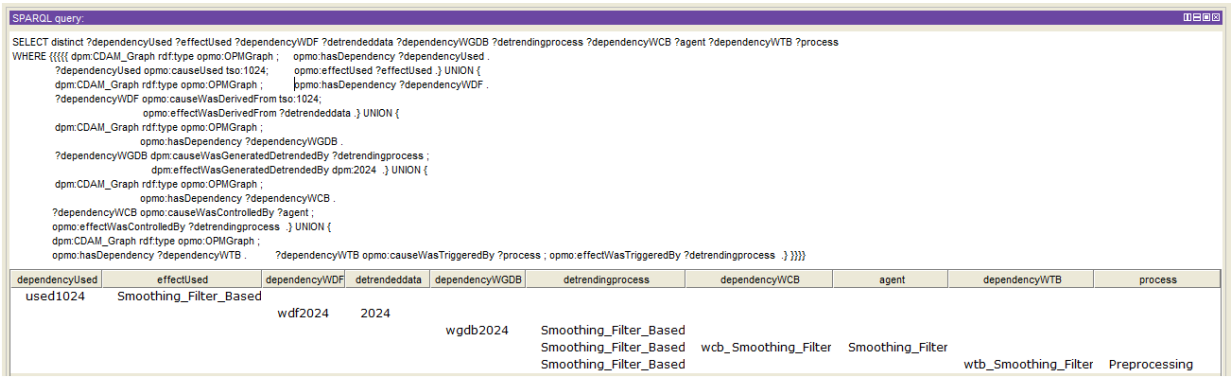


Figura 7.38: Grafo de proveniência de uma série, relações de causa e efeito de suas dependências.

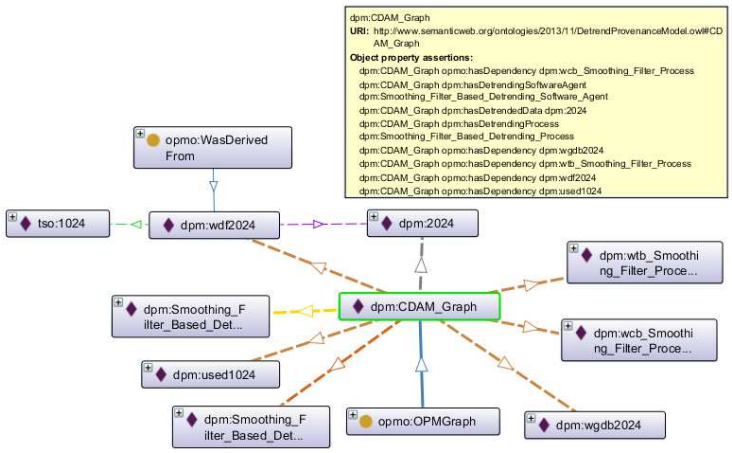
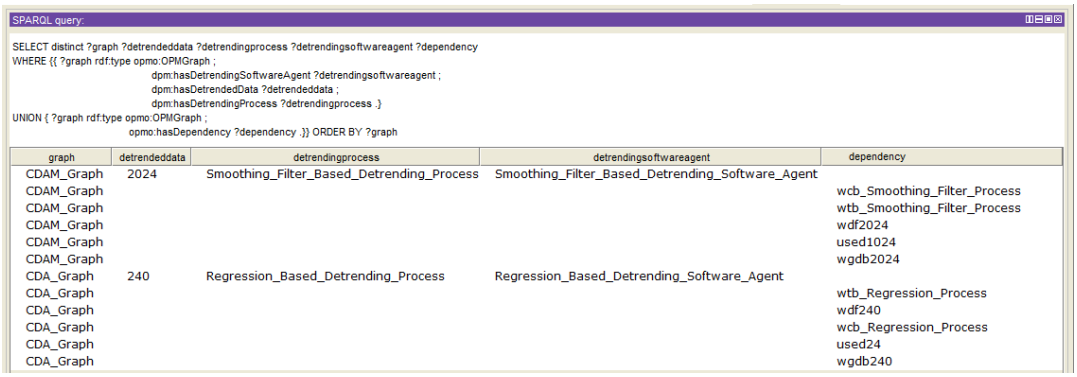


Figura 7.39: Grafo de proveniência de uma série e dependências.

A Figura 7.41 apresenta quais séries temporais foram geradas e *detrended* por quais processos de *detrending* e seus tipos. A Figura 7.42 apresenta o relacionamento WCB,



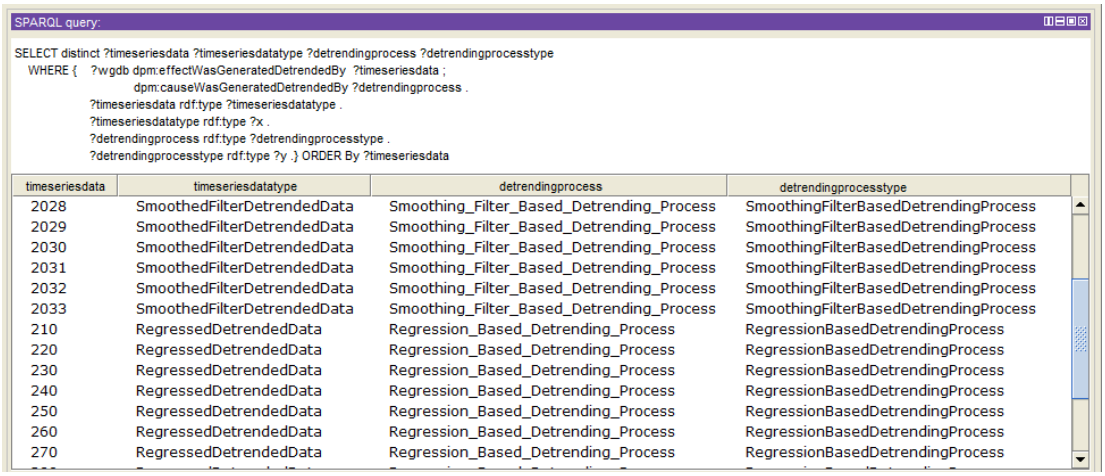
SPARQL query:

```
SELECT distinct ?graph ?detrendeddata ?detrendingprocess ?detrendingsoftwareagent ?dependency
WHERE {{ ?graph rdf:type opmo:OPMGraph ;
                dpm:hasDetrendingSoftwareAgent ?detrendingsoftwareagent ;
                dpm:hasDetrendedData ?detrendeddata ;
                dpm:hasDetrendingProcess ?detrendingprocess .}
UNION { ?graph rdf:type opmo:OPMGraph ;
                opmo:hasDependency ?dependency .}} ORDER BY ?graph
```

graph	detrendeddata	detrendingprocess	detrendingsoftwareagent	dependency
CDAM_Graph	2024	Smoothing_Filter_Based_Detrending_Process	Smoothing_Filter_Based_Detrending_Software_Agent	
CDAM_Graph				wcb_Smoothing_Filter_Process
CDAM_Graph				wtb_Smoothing_Filter_Process
CDAM_Graph				wdf2024
CDAM_Graph				used1024
CDAM_Graph				wgdb2024
CDA_Graph	240	Regression_Based_Detrending_Process	Regression_Based_Detrending_Software_Agent	
CDA_Graph				wtb_Regression_Process
CDA_Graph				wdf240
CDA_Graph				wcb_Regression_Process
CDA_Graph				used24
CDA_Graph				wgdb240

Figura 7.40: Grafos CDA e CDA-M e artefatos, processos e agentes relacionados.

mostrando os agentes que dispararam os processos de *detrending* e quais os respectivos softwares e algoritmos relacionados, mostrando qual o método de *detrending* utilizado e sua aplicabilidade no referido algoritmo. As Figuras 7.43 a 7.46, apresentam, respectivamente, os relacionamentos Used, WCB, WTB e WDF, envolvendo processos e séries temporais de ambos os casos de uso apresentados.



SPARQL query:

```
SELECT distinct ?timeseriesdata ?timeseriesdatatype ?detrendingprocess ?detrendingprocesstype
  WHERE {   ?wgdb dpm:effectWasGeneratedDetrendedBy  ?timeseriesdata ;
                  dpm:causeWasGeneratedDetrendedBy ?detrendingprocess .
            ?timeseriesdata rdf:type ?timeseriesdatatype .
            ?timeseriesdatatype rdf:type ?x .
            ?detrendingprocess rdf:type ?detrendingprocesstype .
            ?detrendingprocesstype rdf:type ?y .} ORDER By ?timeseriesdata
```

timeseriesdata	timeseriesdatatype	detrendingprocess	detrendingprocesstype
2028	SmoothedFilterDetrendedData	Smoothing_Filter_Based_Detrending_Process	SmoothingFilterBasedDetrendingProcess
2029	SmoothedFilterDetrendedData	Smoothing_Filter_Based_Detrending_Process	SmoothingFilterBasedDetrendingProcess
2030	SmoothedFilterDetrendedData	Smoothing_Filter_Based_Detrending_Process	SmoothingFilterBasedDetrendingProcess
2031	SmoothedFilterDetrendedData	Smoothing_Filter_Based_Detrending_Process	SmoothingFilterBasedDetrendingProcess
2032	SmoothedFilterDetrendedData	Smoothing_Filter_Based_Detrending_Process	SmoothingFilterBasedDetrendingProcess
2033	SmoothedFilterDetrendedData	Smoothing_Filter_Based_Detrending_Process	SmoothingFilterBasedDetrendingProcess
210	RegressedDetrendedData	Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess
220	RegressedDetrendedData	Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess
230	RegressedDetrendedData	Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess
240	RegressedDetrendedData	Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess
250	RegressedDetrendedData	Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess
260	RegressedDetrendedData	Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess
270	RegressedDetrendedData	Regression_Based_Detrending_Process	RegressionBasedDetrendingProcess

Figura 7.41: Consulta sobre quais séries foram geradas e corrigidas de tendência e os processos de *detrending* e seus tipos.

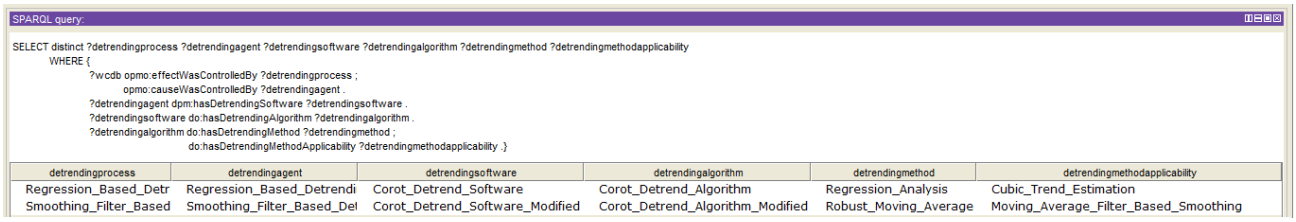


Figura 7.42: Consulta sobre agentes e softwares de *detrending* e algoritmos, incluindo o método e sua aplicabilidade.

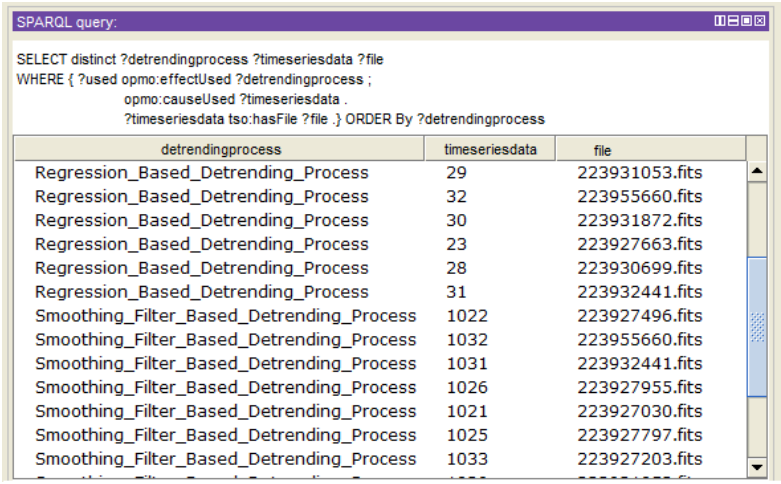


Figura 7.43: Consulta sobre Used, processo de *detrending* e arquivo da série usada.

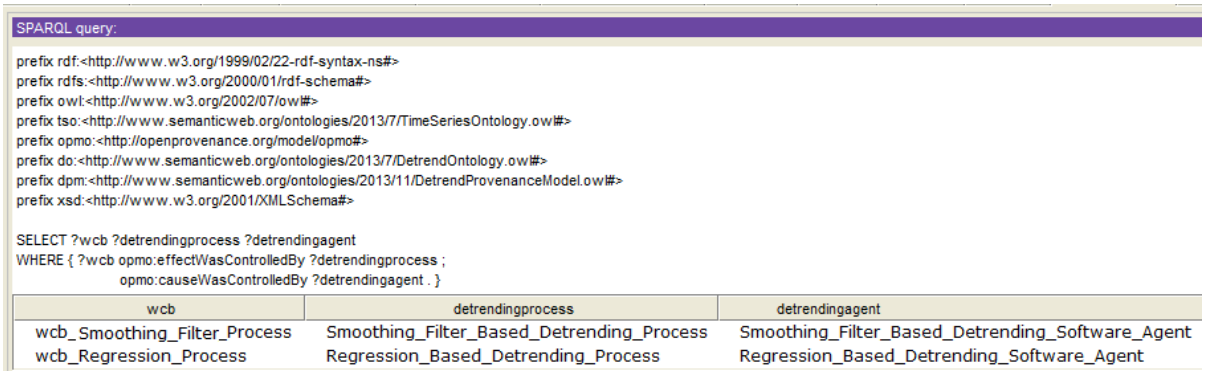


Figura 7.44: Consulta sobre WCB, processo de *detrending* e agente.

SPARQL query:

```
SELECT ?detrendingprocess ?triggeredprocess ?detrendingagent
WHERE { ?wtb opmo:effectWasTriggeredBy ?detrendingprocess ;
          opmo:causeWasTriggeredBy ?triggeredprocess .
        ?wcb opmo:effectWasControlledBy ?detrendingprocess ;
          opmo:causeWasControlledBy ?detrendingagent . }
```

detrendingprocess	triggeredprocess	detrendingagent
Regression_Based_Detrending_Process	Preprocessing	Regression_Based_Detrending_Software_Agent
Smoothing_Filter_Based_Detrending_Process	Preprocessing	Smoothing_Filter_Based_Detrending_Software_Agent

Figura 7.45: Consulta sobre o processo de *detrending*, processo que o disparou e agente.

SPARQL query:

```
SELECT distinct ?detrendeddata ?detrendeddatatype ?timeseriesdata ?timeseriesdatatype
    WHERE { ?wdf opmo:effectWasDerivedFrom ?detrendeddata ;
              opmo:causeWasDerivedFrom ?timeseriesdata .
            ?detrendeddata rdf:type ?detrendeddatatype .
            ?detrendeddatatype rdf:type ?x .
            ?timeseriesdata rdf:type ?timeseriesdatatype .
            ?timeseriesdatatype rdf:type ?y . } ORDER BY ?detrendeddata
```

detrendeddata	detrendeddatatype	timeseriesdata	timeseriesdatatype
2032	SmoothedFilterDetrendedData	1032	RawData
2033	SmoothedFilterDetrendedData	1033	RawData
210	RegressedDetrendedData	21	FitsSample
210	RegressedDetrendedData	21	RawData
220	RegressedDetrendedData	22	FitsSample
220	RegressedDetrendedData	22	RawData
230	RegressedDetrendedData	23	FitsSample
230	RegressedDetrendedData	23	RawData
240	RegressedDetrendedData	24	NonLinearTimeSeries
240	RegressedDetrendedData	24	DiscreteTimeSeries
240	RegressedDetrendedData	24	NonSeasonalTimeSeries
240	RegressedDetrendedData	24	FitsSample
240	RegressedDetrendedData	24	NormalTimeSeries

Figura 7.46: Consulta sobre um dado *detrended* derivado a partir de (tso:TimeSeriesData).

Os dois casos de uso reais apresentados demonstram que, a partir da extensão de OPM, é possível obter informações sobre artefatos, processos e agentes, gerando informações de proveniência sobre as séries temporais e os métodos e sua aplicabilidade nos algoritmos e softwares de *detrending*.

A próxima seção descreve trabalhos correlatos, apresentando uma análise comparativa dos mesmos com a contribuição desta tese.

7.3 Trabalhos Correlatos

Os trabalhos correlatos são descritos na sequência, seguidos por uma análise comparativa considerando critérios relacionados ao modelo definido. Destaca-se que até o momento não foi encontrado trabalho semelhante que modele a proveniência de características intrínsecas das séries temporais, gerando informações de proveniência quanto à aplicabilidade de processos usados para correção de tendências na fase de pré-processamento da análise de séries temporais.

7.3.1 Henson et al (2009)

Henson et al [150] citam que existem muitas maneiras de como sensores geram e representam dados observacionais, incluindo formatos de dados, unidades de medidas, resolução

espaço-temporal, domínio de aplicação, qualidade de observação, frequência, porcentagem de dados ausentes, entre outros, as quais afetam a integração de dados, a partir de diferentes sensores. Ao integrar dados a partir de diferentes fontes ou mapear dados para modelos de informação de sensores, a semântica dos dados necessita ser bem entendida. Neste trabalho, um modelo de observação de séries temporais é um método comum de representar dados de sensores com uma ordem temporal linear. Observações de séries temporais são utilizadas em vários campos, tais como Estatística e Processamento de Sinais para análises avançadas e previsão, conforme [150].

Esse trabalho modela observações de séries temporais em Linguagem de Ontologia da Web (OWL), tomando por base o Modelo de Observações e Medidas (O&M), constituído de modelos-padrão e esquema em Linguagem (*eXtensible Markup Language* - XML), para codificar observações e medidas a partir de sensores, ambos de tempo real ou não. O Modelo O&M é proposto pelo Consórcio Geoespacial Aberto (*Open Geospatial Consortium* - OGC), estabelecido por *Sensor Web Enablement* (SWE), como um conjunto de especificações, modelos de dados e serviços Web, permitindo que sensores sejam acessíveis e controláveis via Web.

O Modelo O&M permite a integração em nível sintático, sendo necessária a integração em nível semântico. É descrito como observações de séries temporais podem ser modeladas em uma ontologia, objetivando resolver problemas relacionados à integração e consultas. Na integração, diferentes redes de sensores podem representar dados de observações de sensores, usando um modelo comum e vários métodos de sensoriamento não explicitamente representados. Quanto às consultas, esses problemas resultam a partir da necessidade de conhecer, à *priori*, o método de sensoriamento usado para gerar um conjunto de dados, para uma interpretação correta dos resultados de uma consulta. Conforme descrito, os problemas podem ser resolvidos com uma descrição semântica das observações das séries temporais, sendo apresentado um modelo de ontologia, com um estudo de caso real, no domínio de gerenciamento de recursos de pluviosidade no Rio Sul Esk na Tasmânia, Austrália.

Existem várias maneiras de monitorar, coletar e representar dados de sensores com observações de séries temporais. No estudo de caso desenvolvido, foram identificados quatro métodos distintos de monitoramento pluviométrico, divididos nas dimensões: Acumulativo, onde sistemas incrementam valores resultantes de observações continuamente, como um progresso de monitoramento através do tempo; Não-Acumulativo, onde sistemas não são incrementais e fornecem um valor independente para cada resultado da observação; Baseado em Intervalo, onde sistemas geram valores resultantes de observações em pontos discretos, dentro de um intervalo especificado de tempo e; Basedo em Evento, onde sistemas geram valores resultantes de observações somente quando ocorreu um evento.

A representação ontológica de observações de séries temporais usa os prefixos (oml-owl:) para identificar a codificação OWL de O&M e inclui três classes: uma classe descrevendo uma observação básica (oml-owl:Observation), uma classe descrevendo uma coleção de observações (oml-owl:ObservationCollection) e uma classe descrevendo uma observação de série temporal (oml-owl:TimeSeriesObservation), cada qual definindo propriedades e restrições de propriedades em termos de domínio e intervalo. Os relacionamentos para a Classe Observação incluem: (oml-owl:featureOfInterest), sendo uma representação do objeto que a observação é feita; (oml-owl:observedProperty), descreve o fenômeno para o qual o resultado da observação fornece uma estimativa de seu valor; (oml-owl:samplingTime) é o tempo quando o fenômeno foi medido; (oml-owl:observationLocation), a localização do evento da observação; (oml-owl:result), uma estimativa do valor de alguma

propriedade gerada; (oml-owl:procedure), uma descrição de um processo usado para gerar o resultado e (owl-owl:memberOf), uma relação de um conjunto de observações para uma coleção. O relacionamento para a classe Coleção inclui o inverso da última relação, (owl-owl:memberOf), a partir de uma coleção para uma observação.

A classe (oml-owl:TimeSeriesObservation) é um tipo especializado de ambas as classes (oml-owl:ObservationCollection) e (oml-owl:Observation), herdando propriedades dessas classes. São criadas a partir desta classe, duas subclasses: (oml-owl:EventBasedTimeSeries-Observation) e (oml-owl:IntervalBasedTimeSeriesObservation), para representar a forma na qual o sistema é baseado e duas sub-propriedades de (oml-owl:samplingTime): (oml-owl:-eventBasedSamplingTime) e (oml-owl:intervalBasedSamplingTime). Dessa forma, a ontologia OWL, desenvolvida com base no Modelo de Organizações e Medidas, permite adicionar valor aos dados de sensores de séries temporais, no contexto de *Semantic Sensor Web*, descrevendo como a linguagem OWL permite modelar melhores restrições nos dados do que usando a linguagem XML, promovendo interoperabilidade semântica. Vale ressaltar que essa ontologia descreve observações e medidas e não modela outras características intrínsecas das séries temporais.

7.3.2 Bozic (2011)

Bozic [80] define o conceito de séries temporais semânticas, onde tecnologias da Web Semântica são combinadas com modelos de processamento de séries temporais, tornando possível seu uso em novas aplicações. Séries temporais fazem parte de nosso cotidiano, onde, de um lado, a questão é como modelar e simular dados de séries temporais contendo semântica e, de outro lado, como tomar decisões acerca dos dados.

Conforme citado, atualmente, sistemas de suporte a decisão e sistemas de processamento de séries temporais apresentam algumas deficiências, tal como a ausência ou não-integração de meta-informação no processamento das séries temporais, onde contextos e conexões podem não ser corretamente reconhecidos, não havendo a possibilidade para vincular ontologias específicas de domínio em tempo de execução, tornando o processamento específico do domínio difícil de implementar [80].

É proposta uma linguagem de processamento genérica, para simulação e modelagem de séries temporais semânticas, a qual dá suporte à séries temporais homogêneas e heterogêneas, objetivando melhorar a capacidade dos sistemas de suporte à decisão. Um processador habilitado semanticamente usa uma ontologia de domínio como entrada, enriquecendo dados de séries temporais com significado. Isto permite o modo automático de cálculos e o suporte à decisão no domínio, baixa probabilidade de falhas, entre outras características. Os modelos fornecidos por esta linguagem podem ser integrados em sistemas de suporte à decisão interativos para usuários finais.

Uma situação analisada nesse trabalho é que existem diferentes usuários, a partir de diferentes domínios, os quais podem estar interessados em uma determinada série temporal. Mas um problema surge quando os usuários estão interessados em somente alguma informação especial a partir das séries temporais. Nesse ponto, o processamento tradicional de séries temporais deixa de ser suficiente. É proposto o uso de séries temporais semânticas, para melhorar as capacidades dos suportes de sistemas a decisão, envolvidos com o processamento das mesmas. A ideia é enriquecer séries temporais com meta-informação, integrar ontologias no processamento e permitir associações entre dados e metadados. Este processo torna possível, dinamicamente, alterar o processamento de séries temporais, a partir de um domínio genérico para um específico, pela adição de uma ontologia ou,

pela alteração do domínio, por substituir uma ontologia durante o tempo de execução. O componente-chave é o processador de séries temporais, habilitado semanticamente, tendo como entrada ontologias específicas do domínio, enriquecendo os dados de séries temporais com significado, contribuindo na melhoria dos sistemas de suporte à decisão.

Dessa forma, são apresentadas as funcionalidades de visualização de etiquetas (*tag clouds*) de recursos selecionados por um usuário e a funcionalidade de filtragem, a qual filtra etiquetas usando valores específicos, tais como nome, recurso etiquetado, tempo, localização, entre outros. A entrada do processador não é um documento e sim uma série temporal processada pela contagem da frequência das etiquetas, gerando uma *tag cloud* para visualizar a importância e o peso das etiquetas.

Considerando a funcionalidade de filtragem, a solução apresentada é uma extensão para o processamento de séries temporais usando ontologias. Um exemplo de uso da Ontologia de Processamento de Séries Temporais Semânticas (*Semantic Time Series Processing Ontology* - prefixo stsp [33]) é a adição de informações sobre diferentes grupos de interesse nas séries temporais, habilitando o processador para fornecer a saída correta para cada domínio.

Na Ontologia STSP, existem as classes stsp:TimeSeries e stsp:InterestedGroup e a propriedade de objeto stsp:isInterestedIn e as instâncias como stsp:Government do tipo stsp:InterestedGroup, com relacionamentos do tipo stsp:isInterestedIn para as instâncias stsp:EnvironmentalTimeSeries e stsp:FinancialTimeSeries, que são do tipo stsp:TimeSeries. O objetivo de visualização e filtragem é visualizar as etiquetas para os usuários e fazer operações de filtragem requisitadas.

A linguagem proposta fornece modelos de séries temporais ambientais, tais como o Modelo Auto-Regressivo, usado para previsão das séries temporais, e o Modelo de Médias Móveis, usado para modelagem de séries temporais univariadas. Esses modelos podem ser dinamicamente alterados e são implementados em Linguagem Python, os quais podem ser estendidos pelos usuários. Dessa forma, esse trabalho propõe uma linguagem genérica de processamento, para simulação e modelagem de séries temporais semânticas, fornecendo informações adicionais, para melhor tomada de decisão considerando não somente os dados, mas também seu contexto. O protótipo dá suporte à visualização de dados de séries temporais semânticas em *tag clouds* e em filtragem, baseadas na meta-informação das séries temporais. A principal vantagem citada é a possibilidade de processar e preparar os mesmos dados de séries temporais para uso em diferentes domínios.

7.3.3 Bozic e Winiwarter (2012 e 2013)

Bozic e Winiwarter [82] apresentam a evolução da linguagem genérica para séries temporais semânticas (*Time Series Semantic Language* - TSSL), para uma linguagem construída pela comunidade interessada em dados de séries temporais, permitindo observar o fluxo de dados, como dados de sensores com informações adicionais, rotulando postagens de cientistas, com um tópico específico de pesquisa. Essa maneira de processar dados de séries temporais permite endereçar e criar comunidades etiquetadas [82].

A linguagem dá suporte ao processamento de séries temporais homogêneas (com grades de tempo fixas) e heterogêneas, as quais podem ter estruturas de dados complexas, sendo possível trabalhar com padrões de tempo, intervalos e com simples *slots*. Algumas expressões comuns que podem ser usadas são apresentadas, mostrando como expressões modificam as séries temporais.

A arquitetura TSSL fornece alto nível de expressividade, sintaxe amigável ao usuário,

extensibilidade, permitindo gerar modelos de dados significantes. Apresenta o Processador de Séries Temporais (TSP), o qual coordena todo o processo de workflow das séries temporais. TSSL não faz nenhuma suposição sobre a proveniência de uma série temporal, somente considera que é um vetor linear de *slots* ordenados cronologicamente. A carga no *slot* tem a forma de pares chave/valor. Chaves são ou simples identificadores ou têm a forma de nomes qualificados, usados para referenciar elementos particulares, dentro de documentos XML ou *Internationalized Resource Identifiers* - IRIs. Valores são ou quaisquer exemplos anteriores ou literais, tais como *strings*, inteiros, *floats*, objetos específicos da aplicação como imagens e matrizes, podendo ter durações de tempo ou padrões de tempo.

Na arquitetura da Web Semântica, TSP é quem faz a comunicação com o *parser* e o interpretador, assim como faz extensões da linguagem e a comunicação com armazéns de dados. Após obter dados das séries e expressões, as mesmas são traduzidas, onde o *parser* necessita entender a semântica na expressão e pode usar, adicionalmente, a extensão do processador semântico para processar a saída. Os dados podem ser armazenados em formato RDF ou em outro armazém de dados. Os resultados são interpretados e retornam ao processador das séries temporais.

Segundo os autores, a evolução do processamento de séries temporais requer a possibilidade de processar não somente dados de séries temporais, mas também seu significado, assim como a possibilidade de gerar novas informações através de *links*, entre diferentes estruturas de dados. O processamento de séries temporais tradicionais contém valores e *time stamps*, onde o processamento de dados é suficiente.

Linguagens de séries temporais em geral tratam unidades, e isto já é considerado uma extensão útil para séries temporais tradicionais, pois o conhecimento das unidades torna possível processar séries temporais com diferentes unidades, tais como centímetros e metros, necessitando de diferentes interpretações. A evolução é adicionar meta-informação para as séries, onde o significado pode ser adicionado aos dados das séries, para conectar valores e unidades, ou para definir o significado de certos termos especialistas. Nesse contexto, a linguagem proposta é capaz de usar ontologias pré-definidas ou não, para adicionar significado aos dados.

Quanto à diferença entre um processador de séries temporais comum, cuja entrada é uma série com *time stamps* e valores, esse fornece resultados que não contêm nenhuma informação sobre o contexto dos dados processados. Nesse caso, os resultados necessitam ser processados novamente, por outro processador semântico, para adicionar significado para os dados de saída. Por outro lado, os autores citam que um processador semanticamente enriquecido já contém toda a informação necessária considerando o significado da saída, sendo diretamente usado no domínio de destino.

Assim, a linguagem TSSP permite fazer cálculos independente de contexto e específicos do domínio. Dados de séries temporais meteorológicas podem ser interessantes para diferentes domínios tais como governo, gerenciamento de eventos, tráfico aéreo, agricultura, turismo, entre outras, porém, todo domínio é interessado em uma visão diferente dos dados. A semântica contribui para fornecer a informação certa para o grupo de interesse certo. Diferentes comunidades científicas têm diferentes interesses, diferentes pontos de vista e diferentes demandas, nas mesmas séries temporais. Nesse caso, é necessário utilizar ontologias específicas de domínio, para fornecer visões dos dados de séries temporais, relevantes para comunidades específicas.

Em contraste, é apresentada uma abordagem de etiquetagem semântica e construção de grupos de interesse, onde a entrada do processador é uma série temporal. Existem

diferentes comunidades que fornecem etiquetas semânticas, cada uma tendo sua própria ontologia. O processador usa a ontologia, juntamente com as etiquetas, para produzir grupos de interesse com séries temporais específicas para cada grupo. O resultado é um número de séries temporais declaradas, para cada grupo de cientistas, com informações específicas do domínio, avaliadas e relevantes para um grupo específico. A principal contribuição deste artigo é o processamento de séries temporais semânticas, relacionado aos dados das séries temporais e seu significado, criando novas informações, por meio de *links* entre diferentes estruturas. Os casos de uso e aplicações estão situados em áreas científicas e a avaliação é feita em projetos tais como Tatoo [42].

Bozic e Winiwarter [83] fazem uma demonstração sobre o processamento de séries temporais semânticas. A funcionalidade da Linguagem Semântica de Séries Temporais (TSSL) é apresentada, assim como outras ferramentas para enriquecer as séries temporais com metadados. Essas tecnologias podem melhorar o processamento de séries temporais, usando uma linguagem dedicada e construída pela comunidade [83].

A aplicabilidade desse artigo é com dados ambientais, medidos a partir de diferentes sensores a serem distribuídos a diferentes grupos de interesse. Os dados são representados como séries temporais de água e qualidade do ar e os grupos de usuários são de agências ambientais, companhias do setor industrial e autoridades. Algumas características são a integração dinâmica de ontologias, anotações de séries temporais e filtragem semântica. Segundo os autores, este trabalho apresenta impacto prático no processamento de séries temporais, fornecendo novas fontes de dados para aplicações da Web Semântica.

7.3.4 Bozic et al (2014)

Bozic et al [81] apresentam um mapeamento de ontologias no processamento de séries temporais semânticas e predição de mudanças climáticas, na forma de um cenário de validação dos métodos para enriquecimento de séries temporais com meta-informação e anotações [81], conforme proposto em Bozic e Winiwarter [82].

As séries temporais envolvem dados a partir de diferentes fontes ou domínios e para seu processamento, existe uma necessidade cada vez maior para endereçar grupos de usuários, interessados em um domínio específico. A aplicação *Climate Twins* é um modelo de predição para geo-regiões, baseadas em temperatura e precipitações. Esta aplicação é usada como cenário para mostrar como selecionar dados de séries temporais adequados e como fornecer recursos adequados para determinados grupos de usuários. Para tanto, é desenvolvida uma ontologia de domínio para um grupo de usuários. A aplicação *Climate Twins* mostra o impacto de mudanças climáticas e permite adaptações para usuários, por encontrar modelos onde o clima futuro de certo ponto de interesse é similar ao clima atual de outra localização, onde essas regiões são chamadas *climate twins* porque apresentam um clima similar.

A ontologia de domínio *Climate Twins* (Ontologia AIT), desenvolvida no Projeto Tatoo, define conceitos de temperatura, precipitação, peso, tempo, como definido nas séries temporais e outros indicadores e parâmetros. O conjunto de Ontologias SWEET [40] e a *Clean Energy Info Portal* (Reegle) [46] têm sido usadas como base para desenvolvimento da ontologia de domínio (prefixo ct:). A ontologia serve de base para *reasoning*, na área de mudanças climáticas, a fim de melhorar a experiência do usuário e gerar conexões entre usuários e os dados (séries temporais). É mostrado nesse trabalho como usuários e dados de séries temporais podem ser unificados em certos grupos de interesse e, portanto, beneficiar usuários pelo recebimento de dados personalizados e o compartilhamento de

tópicos comuns e interesses.

O exemplo apresentado é de um usuário político que trabalha para o Ministério de Meio Ambiente, o qual está interessado em dados de séries temporais sobre mudanças climáticas. O usuário carrega a ontologia de domínio para o sistema e habilita "Mudanças climáticas" como um tópico de interesse, o qual precisa obter dados que dão suporte às suas decisões, para implementar regulamentações e fazer investimentos em determinados campos ambientais. Uma vez selecionado o respectivo tópico de interesse, a partir da ontologia de domínio, é obtido como resultado a lista de séries temporais com esse tópico.

As séries resultantes são recuperadas, baseadas em anotações, a partir de usuários anteriores ou por meio de raciocínio lógico. Nesse caso específico, as séries temporais são recuperadas baseadas na anotação da subclasse Tópico "Mudanças Climáticas", onde todas as séries temporais com este tópico são recuperadas a partir do sistema. Esse é apenas um cenário ilustrado, no qual é feito um mapeamento da ontologia de domínio com a Ontologia Bridge (prefixo bridge:), permitindo o desenvolvimento de regras em Lógica de Descrição, a serem aplicadas por um raciocinador lógico como *Pellet*, permitindo inferências de novos metadados. Com esse cenário, é demonstrado como melhorar a área de predição de mudanças climáticas, processando séries temporais semânticas.

7.3.5 Compton et al (2012)

Compton et al [105] definem a Ontologia *Semantic Sensor Network* (SSN, prefixo ssn:) [32] desenvolvida em Linguagem OWL 2, proposta pelo Grupo Incubador de Rede de Sensores Semânticos W3C (W3C *Semantic Sensor Network Incubator Group*) para descrever sensores em termos de capacidades, processos de medidas, observações e implementações. Sensores são definidos como qualquer coisa que observa, sendo descritos em quaisquer níveis de detalhes, onde humanos e simulações podem ser modelados como sensores [105].

A Iniciativa *Sensor Web Enablement* (SSE), do Consórcio Geoespacial Aberto (OGC), define padronizações para armazenamento e acesso de dados de sensores, como SensorML e o Modelo Observações e Medidas (O&M), fornecendo interoperabilidade sintática, mas necessitando de uma camada semântica. Nesse contexto, tecnologias da Web Semântica permitem interoperabilidade para sensores e sistemas de sensoriamento. A Ontologia SSN cobre grande parte dos padrões SensorML e O&M.

Conforme explicado, o desenvolvimento da ontologia iniciou pela pesquisa de ontologias existentes (doze ontologias foram revisadas) e de padrões, e o desenvolvimento de casos de uso, onde os mesmos foram divididos em quatro categorias principais. Nove das ontologias revisadas cobriam os casos de uso. Discusões entre pequisadores e desenvolvedores revelaram diferentes interpretações de conceitos, mesmo conceitos fundamentais tais como Sensor. Após discussões, o grupo decidiu construir uma ontologia para descrever sensores, compatível com Padrões OGC, sem ser restrita a estes, e, escolhendo a mais ampla definição dos conceitos. Primeiramente, foram definidos os conceitos e relações, depois capacidades de medidas, restrições e implementações. Posteriormente foi feito um alinhamento DOLCEUltralite (DUL, prefixo dul:) [48], por esta ser considerada a mais leve ontologia de fundamentação. O Padrão de Projeto (*Ontology Pattern Design Core Stimulus-Sensor-Observation*) utilizado descreve relações entre sensores, *stimulus* e observações. O alinhamento com a ontologia DUL permitiu melhor compromisso ontológico entre conceitos e relações, restringindo possíveis interpretações dos significados dos conceitos.

A Ontologia SSN apresenta quatro perspectivas: i. Sensor, focando em o que se observa, como observa e o que é observado; ii. Observação, focando em dados e metadados

da observação; iii. Sistema, focando em sistemas de sensores e implementações e; iv. Característica e Propriedade, focando em o que observa uma propriedade particular ou quais observações têm sido feitas sobre uma propriedade.

A ontologia é organizada conceitualmente (não fisicamente) em 10 módulos, contendo 41 conceitos e 39 propriedades de objetos, diretamente herdadas de 11 conceitos e 14 propriedades de objeto DUL. Conceitos e propriedades são comentados com as *tags* (rdfs:comment), (rdfs:isDefinedBy), (rdfs:label), (rdfs:seeAlso) e (dc:source). Essa última *tag* usa os termos SKOS, (skos:exactMatch) e (skos:closeMatch), relacionando conceitos e propriedades para SensorML, O&M e ISO/IEC Guide 99:2007 (Vocabulário Internacional de Metrologia).

A Ontologia SSN é combinada com ontologias para propriedades observadas, tais como SWEET e documentos SWE. É citado a possibilidade de organizar a ontologia em módulos fisicamente separados, por exemplo, um contendo o padrão de projeto, o outro, os conceitos de sensores e um terceiro módulo contendo implementações, onde cada módulo importa os demais.

7.3.6 Llaves e Renschler (2012)

Llaves e Renschler [174] estendem a Ontologia *Semantic Sensor Web* W3C (SSN), permitindo modelar ocorrências espaço-temporais inferidas a partir de observações. Um motor (*engine*) de Processamento de Evento Complexo (CEP) processa séries temporais de observações de sensores e abstrai observações de mais alto nível quando os dados correspondem às descrições de eventos [174].

Técnicas de processamento de eventos permitem definir padrões de eventos, podendo ser usadas para detectar mudanças em dados de geo-sensores, em tempo quase real. Esse artigo descreve o termo Evento como: "Qualquer coisa que acontece ou é contemplada como acontecendo em um instante ou sobre um intervalo de tempo que é relevante para o observador".

O objetivo de usar técnicas de processamento de eventos em dados de geo-sensores é: i. detectar mudanças ou padrões de mudanças em *streams* de dados contínuos; e ii. abstrair entidades de mais alto nível, a partir de simples observações. Um evento é chamado extremo se ele está, a partir da cauda de uma distribuição climatológica, acontecendo somente em cinco porcento ou menos do tempo. Em um contexto mais genérico e geográfico, são ocorrências espaço-temporais raras, não esperadas, ou significantes em termos de mudar a funcionalidade de um sistema.

É comentado nesse trabalho sobre o uso do termo "Evento" em diferentes comunidades. Uma abordagem para analisar dados de geo-sensores fornecidos em tempo real trata toda simples observação como um evento, utilizada em um contexto ontológico. Eventos de observações são eventos geo-espaciais, porque estão situados em um contexto espaço-temporal, mas isto fica implícito no processo de observação.

Segundo os autores, algumas comunidades usam o mesmo termo para categorizar situações ocorrendo sob diferentes condições. Dessa forma, cada organização usa diferentes granularidades e diferentes critérios, os quais podem levar a uma falta de entendimento ao comparar eventos. Assim, o conhecimento sobre as regras de categorização usadas para definir ocorrências ambientais é essencial para integrar dados entre diferentes comunidades.

Para evitar ambiguidades, nesse trabalho, a Ontologia SSN é estendida para modelar ocorrências espaço-temporais inferidas a partir de observações. Um CEP *engine* processa séries temporais de observações de sensores e abstrai o mais alto nível de abstrações,

quando os dados correspondem às descrições de eventos. Na ontologia estendida (prefixo eo:), a detecção de uma mudança em dados de sensores por meio de um processamento de evento, é considerada como uma nova observação. A extensão contempla cinco novos conceitos especializados e uma relação.

Uma classe (eo:EventObservation) é uma situação observada, de mudança na propriedade da entidade geográfica, sendo observada. Uma classe (ssn:Observation) é uma situação que satisfaz uma descrição do método usado para observá-lo. O método de sensoriamento usado é descrito na classe (eo:EventDetectionProcedure) e a descrição da situação em si é definida na classe (eo:EventObservationRule), a qual é uma regra usada para disparar a detecção de mudança, quando o dado analisado corresponde a determinadas condições. Um (eo:EventObservationType) pode estar relacionado a diferentes regras em (eo:EventObservationRules), assim este modelo contribui para resolver o problema de diferentes visões da mesma ocorrência de eventos. A localização espaço-temporal (eo:EventObservation) observada é inferida, a partir das observações de mais baixo nível, e processadas por (eo:EventProcessingAgent). O evento disparando a observação é representado como um (ssn:Stimulus) e sua identidade não é afetada por diferentes visões. Para inferir ocorrências ambientais usando CEP, é necessário converter cada observação individual em um evento.

Essa abordagem se propõe a resolver problemas de interoperabilidade, de diferentes comunidades, usando o mesmo termo para se referir a ocorrências diferentes. Para tanto, é feita a inclusão da descrição da situação, nos modelos da observação, e não somente na descrição do método de sensoriamento usado. Quanto à proveniência, esse modelo pode ser útil para rastrear inferências feitas sobre as observações das séries temporais. A técnica é considerada pelos autores como promissora, existindo a necessidade de mais pesquisas a serem realizadas nesse contexto.

7.3.7 Sheth et al (2008)

Sheth et al [222] descrevem sobre *Semantic Sensor Web* (SSW), onde sensores têm sido incrementalmente adotados para uso em diversas disciplinas, estando distribuídos globalmente, resultando em uma avalanche de dados ambientais. A tecnologia de sensores envolve diferentes tipos de sensores, remotos ou não, com diversas capacidades. Uma rede de sensores possibilita detectar e identificar uma variedade de observações, desde simples fenômenos a eventos complexos. Porém, a falta de integração e comunicação entre essas redes, isola importantes *streams* de dados e intensifica problemas existentes de muitos dados e conhecimento insuficiente.

É discutido sobre *Semantic Sensor Web* (SSW), onde dados de sensores são anotados com metadados semânticos, aumentando a interoperabilidade para fornecer informações contextuais apropriadas, agregando dados de sensores com metadados semânticos espaciais, temporais e temáticos.

A abordagem é baseada no Consórcio Geoespacial Aberto (OGC) e *Semantic Web Activity*, do Consórcio W3C, para fornecer melhores descrições e adicionar significado para dados de sensores. O OGC estabelece o *Sensor Web Enablement* (SWE), para propor um conjunto de especificações relacionadas a sensores, modelos de dados de sensores e serviços de sensores Web, permitindo que sensores sejam acessados e controlados via Web. Dentre as especificações propostas, se destaca o Modelo de Observações e Medidas (O&M), desenvolvido na Linguagem *eXtensible Markup Language* (XML) para codificar observações arquivadas e em tempo real, obtidas a partir de sensores.

No contexto da Web Semântica, a qual adiciona significado à Web tradicional, o

significado da informação é formalmente definido. Definições formais são feitas por ontologias, tornado possível as máquinas interpretarem os dados. As tecnologias relacionadas com a Web Semântica são *Resource Description Framework* (RDF) e *Web Ontology Language* (OWL). A codificação de sensores de fenômenos observados é de natureza opaca (em formatos binários ou proprietários). Segundo os autores, metadados representam um papel essencial para gerenciar dados de sensores. Uma rede de sensores semanticamente enriquecida deveria fornecer informações espaciais, temporais e temáticas, essenciais para descobrir e analisar dados de sensores.

Metadados espaciais fornecem informações, considerando a localização do sensor e dos dados, em termos do sistema de referência geográfica, local ou localização nomeada. Metadados temporais fornecem informações, considerando o instante de tempo ou intervalo que o dado do sensor é capturado. Metadados temáticos podem ser criados ou derivados de várias formas, tais como análise de dados de sensores, extração de descrições textuais ou etiquetagem social. SSW integra metadados semânticos, no domínio de sensores, usando anotações semânticas, e ontologias e regras, as quais possuem um papel relevante para interoperabilidade, análise e raciocínio lógico sobre dados de sensores heterogêneos.

7.3.8 Análise Comparativa

A Tabela 7.2 apresenta um comparativo entre os trabalhos, incluindo os autores; a descrição e a aplicabilidade; a descrição de características das séries temporais; o uso de um modelo de proveniência; quais as características de implementação (ontologias, linguagens, entre outras); a forma de desenvolvimento dos trabalhos; a associação com a base semântica DBpedia; o reuso de recursos (ontologias, entre outros); o alinhamento com uma Ontologia de Fundamentação e o uso de padrões de projeto de ontologias.

Tabela 7.2: Tabela Comparativa - Trabalhos Correlatos. Fonte: Os autores.

Autores	Descrição	Aplicabilidade	Características das séries temporais	Uso de um Modelo de Proveniência	Características da Implementação	Desenvolvimento Modular	Associação com DBpedia	Reuso de Recursos	Alinhamento com Ontologia de Fundamentação	Padrões de Projeto de Ontologias
Henson et al (2009) [150]	Representação ontológica de observações de séries temporais	Observações de Séries Temporais; *Semantic Sensor Web*	Descreve semanticamente observações de séries temporais em linguagem OWL	-	Implementação OWL-DL codificando o Modelo O&M, prefixo (om-owl)	-	-	Representação ontológica baseada no Modelo de Observações e Medidas (O&M) de *Open Geoespatial Consortium/Sensor Web Enablement*	-	-
Bozic (2011)[80]	Simulação e modelagem de séries temporais semanticamente enriquecidas	Processamento de séries temporais (Informática Ambiental, podendo ser estendido para outras áreas) e a Web Semântica	Define séries temporais semânticas, onde tecnologias da Web Semântica são combinadas com modelos de processamento de séries temporais	-	É proposto uma linguagem genérica para processar dados de séries temporais, um processador habilitado semanticamente para usar ontologias de domínio, enriquecendo as séries temporais com significado apropriado	-	-	-	-	-
Bozic e Winiwarter (2012)[82]	Enriquecimento semântico de séries temporais e o uso da tecnologia para construir vários tipos de comunidades interessadas em dados de séries temporais	Processamento de séries temporais e a Web Semântica	Demonstra uma nova maneira de processar séries temporais semânticas usando a Linguagem Semântica de Séries Temporais (TSSL)	-	A Linguagem Semântica de Séries Temporais (TSSL) é estendida para muitos campos de aplicação, sendo construída pela comunidade	-	-	-	-	-
Bozic e Winiwarter (2013)[83]	Demonstra o processamento de séries temporais semânticas	Processamento de séries temporais ambientais e a Web Semântica, podendo ser estendido para plataformas Web Social	Relaciona-se com o processamento de séries temporais e fornece novas fontes de dados para aplicações da Web Semântica	-	É demonstrada a linguagem semântica de séries temporais (TSSL). Uma ontologia para validação é projetada e usada uma base de conhecimento para armazenamento dos dados	-	-	-	-	-

Tabela 7.3: Tabela Comparativa - Trabalhos Correlatos - Cont.

Autores	Descrição	Aplicabilidade	Características das séries temporais	Uso de um Modelo de Proveniência	Características da Implementação	Desenvolvimento Modular	Associação com DBpedia	Reuso de Recursos	Alinhamento Ontologia de Fundamentação	Padrões de Projeto de Ontologias
Bozic et al (2014)[81]	Apresenta uma validação dos métodos de Bozic et al (2012) em caso de uso de predição de mudanças climáticas com o desenvolvimento da Ontologia de domínio *Climate Twins* (Ontologia AIT)	Processamento de séries temporais semânticas e predição de mudanças climáticas (Projeto Tatoo)	A ontologia AIT serve de base para *reasoning* na área de mudanças climáticas, a fim de melhorar a experiência do usuário e gerar conexões entre usuários e os dados (séries temporais)	-	A Ontologia AIT propõe melhorias na área de predição de mudanças climáticas com processamento de séries temporais semânticas	-	-	Desenvolvimento da Ontologia AIT com base nas Ontologias SWEET e a *Clean Energy Info Portal*	-	-
Compton et al (2012)[105]	Descrição ontológica de sensores considerando capacidades, processos de medida, observações e implementações	Ontologia usada em aplicações para gerenciamento ambiental	A Ontologia SSN tem 4 perspectivas: i. sensor; ii. observação; iii. sistema e iv. característica e propriedade	-	Ontologia OWL2, conceitos e propriedades anotados com (rdfs:comment), (rdfs:isDefinedBy), (rdfs:label), (rdfs:seeAlso) e (dc:source), relacionados com termos SKOS. Esssa ontologia cobre parte dos padrões SensorML e O&M.	A Ontologia SSN é organizada conceitualmente (não fisicamente) em 10 módulos, contendo 41 conceitos e 39 propriedades de objetos, herdados de 11 conceitos DUL e 14 propriedades de objeto DUL	O projeto SPITFIRE combina a ontologia com um modelo de evento, um contexto e uma ontologia de cognição e *linked geodata* e DBpedia	A Ontologia SSN é combinada com Ontologias SWEET, dados de mapeamento United Kingdom e Ontologias Sensorgrid4env	Alinhamento *DOLCE-UltraLite* (DUL)	Padrão de Observação (*Core Stimulus-Sensor-Observation*) - (*design pattern SSO*)
Llaves e Renschler (2012)[174]	Observação de mudanças em observações de sensores em tempo real	*Semantic Sensor Web*; Processamento de Eventos	A Ontologia SSN é estendida para modelar ocorrências espaço-temporais	Este modelo pode ser usado para rastrear informações feitas sobre as séries temporais	A Ontologia SSN é estendida, consistindo de 5 novos conceitos especializados e uma nova relação. O foco é detecção de mudanças em observações de séries temporais	-	-	Herdado de SSN	Alinhamento *DOLCE-UltraLite* (DUL)	Herdado de SSN
Sheth et al (2008)[222]	*Semantic Sensor Web* (SSW)	Semântica de Sensores	SSW permite interoperabilidade e análises avançadas de sensores heterogêneos	-	Semântica de sensores dentro de Espaço, Tempo e Tema	Semantic Sensor Web (SSW) representado como um conjunto de ontologias: temporal, geoespacial, sensor e tempo	-	Modelo de Observações e Medidas (O&M) (entre outros) de *Open Geoespatial Consortium/Sensor Web Enablement*	-	-

Ao analisar os trabalhos correlatos, é possível concluir que:

- Observações de séries temporais podem ser descritas usando a Linguagem OWL, onde ontologias de domínio são utilizadas para enriquecer semanticamente as séries temporais.
- Um dos padrões que se destaca para uso como base para descrever Observações de Séries Temporais é o Modelo Organizações e Medidas (O&M), proposto pelo Consórcio OGC/SWE, o qual apresenta interoperabilidade sintática, mas precisa ser adaptado para a Linguagem OWL, para permitir interoperabilidade semântica.
- É possível associar as necessidades dos usuários com os dados (séries temporais), como forma de contribuir para a tomada de decisão pelos usuários.
- Em geral, os trabalhos analisados de observações de séries temporais não tem como foco principal a proveniência das séries temporais, e não fazem uso de um modelo de proveniência.
- Em séries temporais ambientais, é identificado que existem diferentes interpretações de conceitos, inclusive de conceitos fundamentais, tais como Sensor. Nesse caso, é geralmente escolhida a definição mais ampla para que a mesma seja especializada em sub-conceitos.
- O vocabulário *Simple Knowledge Organization System Reference* (SKOS) [51] pode ser utilizado para definição dos termos, usando a *tag* (dc:source), conforme a Ontologia SSN.
- Termos podem ser associados a *Linked Sensor Data* e *Linked Open Data*, para interoperabilidade semântica.
- Ontologias SWEET são reutilizadas como um padrão de fato, para representar o domínio de ciências ambientais e da terra, conforme analisado por [116].
- O desenvolvimento modular pode ser considerado, como no caso da Ontologia SSN, organizada conceitualmente em dez módulos, a qual pode ser mapeada para uma modularização física.
- Quando é feito um mapeamento para uma Ontologia de Fundamentação, a ontologia considerada é a DUL, devido à sua característica de ser uma ontologia leve, considerando as demais Ontologias de Fundamentação existentes.
- *Ontology Design Patterns* (ODPs) são considerados para o projeto de ontologias de domínio.
- Podem haver problemas de interoperabilidade de diferentes comunidades, usando o mesmo termo, para se referir a diferentes ocorrências e, esse tipo de problema, deve ser tratado semanticamente.

Analisando os trabalhos correlatos, assim como os resultados de consultas em navegadores semânticos, não foi encontrado até o momento, uma ontologia que modele características intrínsecas das séries temporais, métodos usados para *detrending* e a proveniência sobre como as séries temporais foram corrigidas quanto à extração do componente tendência.

Dentre os trabalhos correlatos, destaca-se o trabalho de Bozic et al (2014) [81], o qual objetiva a predição de séries temporais climáticas, enriquecidas semanticamente por meio de uma ontologia. É feita a associação das séries temporais com os objetivos declarados pelo usuário, relacionando quais séries temporais estão associadas com determinado tópico, apresentando essa associação aos usuários [81].

Na ontologia definida nesta tese, informações de proveniência quanto as características intrínsecas das séries temporais são associadas às mesmas, as quais são consideradas informações relevantes, por parte dos pesquisadores, pois facilitam a tomada de decisão sobre qual método utilizar, para determinado tipo de série temporal, em um processo de *detrending*. Da mesma forma, o conhecimento sobre os métodos facilita a tomada de decisão quanto à quais séries temporais os mesmos podem ser aplicados ou podem produzir melhores resultados.

Características consideradas relevantes e adequadas no desenvolvimento das ontologias foram identificadas e consideradas na definição do modelo de proveniência, tais como: i. desenvolvimento modular; ii. definição dos conceitos e propriedades usando as *tags* (rdfs:comment) e (rdfs:label); associação das instâncias com a DBpedia; reuso de declarações semânticas a partir da Ontologia SWEET, e reuso de ontologias e de um modelo de proveniência.

Destaca-se que os módulos ontológicos propostos nesta tese podem ser estendidos e contribuem para o pesquisador conhecer e entender melhor as características dos dados (séries temporais), facilitando a tomada de decisão sobre como tornar as séries temporais estacionárias, melhorando semanticamente um relevante passo da fase de pré-processamento da análise de séries temporais, que é a extração de tendências.

O próximo capítulo apresenta as conclusões e perspectivas de pesquisas futuras.

CAPÍTULO 8

CONCLUSÕES E PERSPECTIVAS DE PESQUISAS FUTURAS

8.1 Conclusões

Séries temporais são observações de dados sobre o tempo, obtidas geralmente em intervalos regulares, em muitas áreas do conhecimento. Sua análise difere da análise de dados tradicional, dada a dependência serial, onde a ordem das observações é relevante na análise. Na fase de pré-processamento, são necessárias correções nas séries para remoção de fenômenos que ocorrem ao longo do tempo, tais como tendências, as quais precisam ser extraídas pois podem ocultar outros fenômenos.

O conhecimento sobre como e com que frequência as séries temporais foram corrigidas, é relevante para a tomada de decisão em um processo de análise. Entretanto, informações sobre as séries temporais e como tendências foram extraídas das mesmas, nem sempre são explícitas e fáceis de interpretar. Nesse contexto, informações de proveniência contribuem para o conhecimento dos dados e dos métodos estatísticos aplicados para sua correção.

Informações de proveniência podem ser obtidas com o uso de metadados, os quais permitem sua descrição. Essa forma de geração de proveniência é de texto livre, podendo gerar ambiguidades. O uso de ontologias para geração de proveniência contribui para a definição de um vocabulário controlado, possibilitando a troca de informações entre humanos e agentes de software e a inferência de conhecimento.

Esta tese é baseada na intersecção dos seguintes tópicos-chave de pesquisa: séries temporais, proveniência, ontologias e a Web Semântica. A principal contribuição é a definição de um modelo usando ontologias para a geração de informações de proveniência sobre as séries temporais e os métodos aplicados para extração de tendência. O modelo é implementado em linguagem OWL, apresentando um projeto modular e centrado no reuso de ontologias e de declarações semânticas, a partir do conjunto de ontologias SWEET. O projeto modular contribui para diminuir a complexidade de modelagem, facilitando discussões e avaliações individuais, promovendo interoperabilidade, reusabilidade, adaptabilidade e extensibilidade das ontologias. A modularidade envolve três ontologias, *Time Series Ontology* - TSO, *Detrend Ontology* - DO e *Detrend Provenance Model* - DPM.

A TSO permite a geração de informações de proveniência nas séries temporais, descrevendo suas características, suposições, modelos de decomposição e componentes. Para o desenvolvimento desta ontologia, as questões de competência foram definidas usando expressões do Modelo conceitual W7, o qual contribuiu para a obtenção do conhecimento quanto a proveniência dos dados. A definição de regras possibilita inferir conhecimento semântico sobre as séries temporais.

A DO adiciona conhecimento semântico quanto aos métodos usados pelos algoritmos para extração de tendências em séries temporais no domínio do tempo, podendo ser estendida para o domínio da frequência. A DPM reutiliza o modelo *Open Provenance Model* - OPM, o qual é estendido para modelar artefatos (séries temporais), processos e agentes de *detrending*. O modelo DPM, definido nesta tese, contribui no que segue:

- Para a geração de informações de proveniência sobre as séries temporais, métodos e processos de *detrending* executados.

- Para o entendimento dos conceitos modelados e de sua proveniência, por meio da definição de classes utilizando bibliografias da análise de séries temporais, as quais podem ser visualizadas por meio de uma documentação online.

- Para interoperabilidade semântica, pelo reuso de ontologias e do modelo OPM, de declarações do conjunto de ontologias SWEET, assim como para interoperabilidade com dados abertos *linkados* (*Linked Open Data*) por meio da associação, quando aplicável, de instâncias com a base de dados semântica DBpedia, permitindo estender as definições da ontologia com demais *tags*, gerando conhecimento.

- Para a reusabilidade das ontologias individuais, para adaptabilidade conforme determinada área do conhecimento e para extensibilidade a partir de declarações reutilizadas, assim como pelo projeto modular.

- Para o entendimento sobre a aplicabilidade dos métodos nas séries temporais. Na ontologia DO, é possível declarar qual o método utilizado e sua aplicabilidade no referido algoritmo. Isso devido ao fato de que o mesmo método pode ser utilizado para realizar mais de uma operação na fase de pré-processamento. Por exemplo, o mesmo método pode ser utilizado tanto para *detrending* quanto para *denoising*, dependendo do objetivo da análise e qual componente da série temporal está sendo analisado.

- Para o entendimento sobre como as séries temporais foram geradas e *detrended*, gerando proveniência sobre as séries temporais originais e quais softwares, métodos e parâmetros foram utilizados na extração de tendências.

- Para o enriquecimento do passo de *detrending*, contribuindo para a melhoria do processo de análise, pois a geração de informações de proveniência e de inferências sobre as séries temporais e o componente tendência permite a escolha de métodos de *detrending* que se aplicam conforme suas características.

- Para a tomada de decisão em um processo de análise, enriquecendo semanticamente um relevante passo da análise de séries temporais, permitindo a escolha de outros métodos que se aplicam nas respectivas séries temporais conforme suas características, os quais podem ser aplicados como forma de se obter melhores resultados, contribuindo para a geração do conhecimento.

Como forma de validação do modelo definido nesta tese, um estudo de caso foi desenvolvido utilizando arquivos do tipo FITS, os quais constituem séries temporais não-estacionárias, necessitando de correção de tendências. Com o Modelo de Proveniência *Detrend*, é possível obter informações quanto à proveniência das séries temporais, os métodos e sua aplicabilidade nos respectivos algoritmos e softwares, os agentes e processos de *detrending* executados. Dessa forma, contribui-se para enriquecer semanticamente um passo do pré-processamento, o qual é necessário em muitas áreas do conhecimento que analisam séries temporais.

Uma questão a considerar quanto ao uso do modelo é o tempo para inferências, o qual pode ser melhorado com o desenvolvimento de raciocinadores lógicos, assim como pela evolução tecnológica. Contudo, apesar do tempo de processamento relacionado à combinação das ontologias, foi apresentado nesta tese que o modelo definido baseado em ontologias pode ser usado efetivamente para geração de informações de proveniência

em séries temporais, enriquecendo o passo de *detrending*, possibilitando a geração de inferências, favorecendo o desenvolvimento de consultas ricas semanticamente em um passo considerado relevante na análise de séries temporais, contribuindo com a geração do conhecimento. Como contribuição científica desta tese, foram apresentados e publicados os seguintes artigos em congressos internacionais:

De SOUZA, Lucélia; VAZ, Maria Salete Marcon Gomes; SUNYE, Marcos Sfair. Modular Development of Ontologies for Provenance in Detrending Time Series. In: Eleventh International Conference on Information Technology: New Generations (ITNG), 2014, Las Vegas. p. 567-572.

De SOUZA, Lucélia; VAZ, Maria Salete Marcon Gomes; SUNYE, Marcos Sfair. Domain Ontology for Time Series Provenance. In: 16th International Conference on Enterprise Information Systems, ICEIS 2014, Lisbon. p. 217-224.

8.2 Perspectivas de Pesquisas Futuras

Como perspectivas de pesquisas futuras, destacam-se:

- O desenvolvimento de um ambiente online baseado no modelo de proveniência definido, contribuindo para facilitar e enriquecer semanticamente o passo de extração de tendências.
- Na ontologia para proveniência de séries temporais, uma sugestão é considerar quando uma mesma série foi alterada quanto ao intervalo de observação.
- Realizar a aplicabilidade da ontologia TSO em outros domínios do conhecimento.
- Na ontologia *Detrend*, métodos usados para extração de tendências no domínio da frequência podem ser adicionados, tais como *wavelets*, assim como a ontologia pode ser estendida para modelar métodos da Estatística Bayesiana.
- A ontologia do modelo de proveniência DPM pode ser especializada para modelar outros métodos de *detrending*, como no caso de métodos no domínio da frequência.
- Como adição de semântica, os elementos das ontologias podem ser associados por meio da *tag* (dc:source) ao vocabulário SKOS, assim como a *Linked Sensor Data*, promovendo interoperabilidade.
- Para descrever as publicações dos algoritmos e/ou softwares de *detreding*, sugere-se reutilizar, como um todo ou uma parte, a ontologia *Semantic Web for Research Communities* - SWRC [34], a qual modela entidades de comunidades de pesquisa, tais como pessoas, organizações, publicações e seus relacionamentos, estando disponível na Web.
- As ontologias podem ser mapeadas a uma ontologia de fundamentação. Sugere-se que a ontologia considerada seja a DUL, por ser uma ontologia utilizada por trabalhos correlatos assim como ser considerada uma ontologia leve ao ser comparada com outras ontologias de fundamentação.
- Quanto ao estudo de caso desenvolvido com séries fotométricas reais, uma sugestão é a adição de metadados gerados, a partir da definição dos elementos das ontologias, no

cabeçalho (*header*) dos arquivos FITS. Isso contribui para gerar informações sobre as séries temporais e como as mesmas foram corrigidas (*detrended*). Para exemplificar, é possível adicionar metadados referentes ao tipo da série temporal, ao tipo de tendência considerada e o método aplicado, entre outras informações, permitindo uma melhor interpretação sobre os procedimentos aplicados e facilitando a escolha de outros métodos de *detrending* que podem ser usados para se obter melhores resultados quanto a correção de tendências.

- Uma sugestão de extensão do modelo de proveniência é quanto à associação dos métodos que são aplicáveis a determinados tipos de séries temporais, assim como em relação a, dada uma série temporal e suas características, quais métodos podem ser utilizados para sua correção. Para exemplificar, o método *Empirical Mode Decomposition* é aplicável para séries temporais não-estacionárias e não-lineares.

REFERÊNCIAS

[1] Ontology matching. URL:http://www.ontologymatching.org/, Dec. 2012.

[2] Plugin owl2 query tab. URL:http://krizik.felk.cvut.cz/km/owl2query/index.html, Dec. 2012.

[3] Abs software. URL:www.abs.gov.au/, Oct. 2013.

[4] Acm ontologies. URL:http://acm.rkbexplorer.com/ontologies/, Aug. 2013.

[5] Center space software. URL:http://www.centerspace.net/blog/savitzky-golay-smoothing/, Sept. 2013.

[6] Convenction, rotation and planetary transits. corot. URL:http://idoc-corot.ias.u-psud.fr/, Aug. 2013.

[7] Core software ontology. URL:http://cos.ontoware.org/cso, July 2013.

[8] Corot archive. URL:http://idoc-corotn2-public.ias.u-psud.fr/, Aug. 2013.

[9] Dcmi type vocabulary. URL:http://dublincore.org/documents/2000/07/11/dcmi-type-vocabulary/, Oct. 2013.

[10] Evoont - software evolution ontology. URL:https://files.ifi.uzh.ch/ddis/oldweb/ddis/research/evoont/, June 2013.

[11] Friend of a friend. URL:http://www.foaf-project.org/, June 2013.

[12] Georss. URL:http://georss.org/, May 2013.

[13] Glossary of statistical terms (oecd). URL:http://stats.oecd.org/glossary/index.htm, Aug. 2013.

[14] International statistical institute. URL:http://isi.cbs.nl/glossary/bloken00.htm, Sept. 2013.

[15] Java file libraries. fits jar. URL:http://www.java2s.com/Code/Jar/f/Downloadfitsjar.htm, July 2013.

[16] Jena framework. URL:http://jena.apache.org/, Mar. 2013.

[17] The mathworks. URL:http://www.mathworks.com/, Oct. 2013.

[18] Norma brasileira abnt nbr iso 3534-1. 1a. ed. 2006 - 2010. URL:http://www.abntcatalogo.com.br/, Oct. 2013.

[19] Normas iso (3534-1:2006, 3534-2:2006 e 3534-3:1999)). URL:http://www.iso.org/iso/home/, Oct. 2013.

[20] Ontograf. URL:http://protegewiki.stanford.edu/wiki/OntoGraf, Sept. 2013.

[21] Ontology computer science for non-computer scientists. project lt4el (csncsv0.01lex). URL:http://watson.kmi.open.ac.uk/ontologies/LT4eL/CSnCSv0.01Lex.owl, June 2013.

[22] Ontosearch. URL:http://www.ontosearch.com/, Dec. 2013.

[23] Open provenance model ontology. URL:http://openprovenance.org/model/opmo, Sept. 2013.

[24] Open provenance model vocabulary. URL:http://purl.org/net/opmv/ns, Sept. 2013.

[25] Owl api. URL:http://owlapi.sourceforge.net/2.x.x/index.html, Mar. 2013.

[26] Oxford dictionary. URL:www.oed.com, Apr. 2013.

[27] Protégé-owl api. URL:http://protege.stanford.edu/plugins/owl/api/, Apr. 2013.

[28] Provenance vocabulary core ontology. URL:http://trdf.sourceforge.net/provenance/ns.html, June 2013.

[29] Reprmathstatistics ontology. URL:http://sweet.jpl.nasa.gov/2.3/reprMathStatistics.owl, Sept. 2013.

[30] Sas stat software. URL:http://www.sas.com/, Oct. 2013.

[31] Sciflow a scientific workflow ontology. URL:http://www.lbd.dcc.ufmg.br/colecoes/sbbd/2011/0016.pdf, Oct. 2013.

[32] Semantic sensor network. URL:http://purl.oclc.NET/ssnx/ssn, Feb. 2013.

[33] Semantic time series processing ontology. URL:http://www.semantic-time-series.org/stsp.owl, Sept. 2013.

[34] Semantic web for research communities. URL:http://ontoware.org/swrc/, Oct. 2013.

[35] Seon. software evolution ontologies. URL:http://www.se-on.org/, July 2013.

[36] Statistical analysis ontology. URL:http://a.com/StatisticalAnalysis.owl, June 2013.

[37] The statistical core vocabulary (scovo). URL:http://sw.joanneum.at/scovo/schema.html, Aug. 2013.

[38] Statistical techniques in the data library: A tutorial. URL:http://iridl.ldeo.columbia.edu/dochelp/StatTutorial/, Sept. 2013.

[39] Statistics glossary. URL:http://www.stats.gla.ac.uk/steps/glossary/paired_data.html, Oct. 2013.

[40] Sweet ontologies. URL:http://sweet.jpl.nasa.gov/, July 2013.

[41] Swo ontology. URL:http://purl.bioontology.org/ontology/SWO, June 2013.

[42] Tatoo project. URL:http://www.tatoo-fp7.eu/tatooweb/, Feb. 2013.

[43] Taxonomia acm. URL:http://dl.acm.org/, Nov. 2013.

[44] Terminology on statistical metadata. URL:http://www.unece.org/fileadmin/DAM/stats/publications/53metadaterminology.pdf, July 2013.

[45] Web of trust. URL:http://xmlns.com/wot/0.1/, May 2013.

[46] Clean energy info portal. URL:http://reegle.info/, Feb. 2014.

[47] Dbpedia. URL:http://dbpedia.org/, Jan. 2014.

[48] Dul ontology. URL:http://www.loa-cnr.it/ontologies/DUL.owl, Feb. 2014.

[49] Ecoinformatics. URL:https://code.ecoinformatics.org/code/wow/trunk/data/OWL/, Jan. 2014.

[50] Provxg. URL:http://www.w3.org/2005/Incubator/prov/wiki/Main_Page, Jan. 2014.

[51] Simple knowledge organization system reference. skos-reference. URL:http://www.w3c.org/TR/skos-reference, Mar. 2014.

[52] Sindice. URL:http://sindice.com, Jan. 2014.

[53] Swoogle. URL:http://swoogle.umbc.edu/, Jan. 2014.

[54] Tupelo project. URL:http://tupeloproject.ncsa.uiuc.edu/node/2, Feb. 2014.

[55] Watson. URL:http://watson.kmi.open.ac.uk/, Feb. 2014.

[56] Watson plugin. URL:http://neon-toolkit.org/wiki/Watson/for/Knowledge/Reuse, Jan. 2014.

[57] Watson wui. URL:http://watson.kmi.open.ac.uk/WatsonWUI/, Jan. 2014.

[58] Alexandrov, T. A method of trend extraction using singular spectrum analysis. *REVSTAT. Statistical Journal 7*, 1 (2009).

[59] Alexandrov, T., Bianconcini, S., Dagum, E. B., Maass, P., and McElroy, T. S. A review of some modern approaches to the problem of trend extraction. *Econometric Reviews 31*, 6 (Nov. 2012), 593–624.

[60] Alexe, B., Chiticariu, L., and Tan, W. Spider: A schema mapping debugger. In *Proceedings of the 32nd international conference on Very large data bases* (2006), VLDB '06, VLDB Endowment, pp. 1179–1182.

[61] Altintas, I., Barney, O., and Jaeger-Frank, E. Provenance collection support in the kepler scientific workflow system. In *IPAW* (2006), L. Moreau and I. T. Foster, Eds., vol. 4145 of *Lecture Notes in Computer Science*, Springer, pp. 118–132.

[62] Altman, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician 46*, 3 (1992).

[63] Anand, M. K., Bowers, S., McPhillips, T., and Ludascher, B. Efficient provenance storage over nested data collections. In *EDBT '09: Proceedings of the 12th International Conference on Extending Database Technology* (New York, NY, USA, 2009), ACM, pp. 958–969.

[64] Andrews, T., and et al. *BPEL4WS, Business Process Execution Language for Web Services Version 1.1.* IBM, 2003.

[65] Artz, D., and Gil, Y. A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web 5*, 2 (June 2007), 58–71.

[66] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, New York, NY, USA, 2003.

[67] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F. *The Description Logic Handbook.* Cambridge University Press, New York, NY, USA, 2007.

[68] Barga, R. S., and Digiampietri, L. A. Automatic capture and efficient storage of e-science experiment provenance. *Concurr. Comput.: Pract. Exper. 20*, 5 (Apr. 2008), 419–429.

[69] Becker, R. A., and Chambers, J. M. Auditing of data analysis. *SSDBM86: Proceedings of the 3rd International Workshop on Statistical and Scientific Database Management* (1986), 78–80.

[70] Bendat, J. S., and Piersol, A. G. *Random Data: Analysis and Measurement Procedures*, 2nd ed. John Wiley and Sons, Inc., New York, USA, 1986.

[71] Benedetti, J. K. On the nonparametric estimation of regression functions. *Journal of Royal Statistical Society Ser. B, 39* (1977).

[72] Benjamin, B., Tung-Lam, D., Jean-Charles, R., and Thierry, R. Trend filtering methods for momentum strategies. *Lyxor Asset Management* (2011).

[73] Berners-Lee, T., Hendler, J., and Lassila, O. The semantic web. *Scientific American 284*, 5 (May 2001), 34–43.

[74] Bhagwat, D., Chiticariu, L., Tan, W., and Vijayvargiya, G. An annotation management system for relational databases. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30* (2004), VLDB '04, VLDB Endowment, pp. 900–911.

[75] Bizer, C., and Berners-Lee, T. Linked data - the story so far. In *Int J Semant Web Inf Syst 5 (3)* (2009), pp. 1–22.

[76] Borst, W. N. *Construction of Engineering Ontologies.* PhD thesis, University of Tweenty Centre for Telematica and Information Technology, Enschede, Nederland, 1997.

[77] Bose, R., and Frew, J. Lineage retrieval for scientific data processing: a survey. *ACM Comput. Surv. 37*, 1 (Mar. 2005), 1–28.

[78] Boufleur, R. C. A busca de exoplanetas com as curvas de luz do corot. Dissertação de mestrado, Departamento de Pós-Graduação. Curso de Pós-Graduação em Astronomia, Rio de Janeiro, Brasil, 2012.

[79] Box, G., and Jenkins, G. *Time Series Analysis, Forecasting and Control*. San Francisco, CA: Holden Day, 1970.

[80] Bozic, B. Simulation and modeling of semantically enriched time series. *19th International Congress on Modelling and Simulation* (2011), 12–16.

[81] Bozic, B., Peters-Andersa, J., and Schimaka, G. Ontology mapping in semantic time series processing and climate change prediction. *International Environmental Modelling and Software Society (iEMSs) 7th International Congress on Environmental Modelling and Software* (2014).

[82] Bozic, B., and Winiwarter, W. Community building based on semantic time series. *iiWAS* (2012), 213–222.

[83] Bozic, B., and Winiwarter, W. A showcase of semantic time series processing. *IJWIS 9*, 2 (2013), 117–141.

[84] Bueno, D. L. S. R. *Econometria de séries temporais*. CENGAGE Learning, 2008.

[85] Buneman, P., Chapman, A., and Cheney, J. Provenance management in curated databases. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 2006), SIGMOD 06, ACM, pp. 539–550.

[86] Buneman, P., Cheney, J., and Vansummeren, S. On the expressiveness of implicit provenance in query and update languages. *ACM Trans. Database Syst. 33*, 4 (Dec. 2008), 1–47.

[87] Buneman, P., Khanna, S., and Tan, W. C. Why and where: A characterization of data provenance. In *ICDT* (2001), pp. 316–330.

[88] Buneman, P., and Tan, W. Archiving scientific data. In *ACM SIGMOD* (2002), pp. 1–12.

[89] Buneman, P., and Tan, W. Provenance in databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 2007), SIGMOD '07, ACM, pp. 1171–1173.

[90] Burke, O. More notes for least squares. Tech. rep., 2010. University of Oxford.

[91] Bussab, W. O., and Morettin, P. A. *Estatística Básica*, 5th ed. ed. Saraiva, 2006.

[92] Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., and Vo, H. T. Vistrails: Visualization meets data management. In *SIGMOD Conference* (2006), pp. 745–747.

[93] Cambrésy, L., Derriere, S., Padovani, P., A., P. M., and Richard, A. Ontology of astronomical object types. Tech. Rep. Version 1.3, Jan. 2010. IVOA Technical Note.

[94] Castleman, K. R. *Digital Image Processing*. Prentice Hall Professional Technical Reference, 1996.

[95] Chandler, R., and Scott, M. *Statistical Methods for Trend Detection and Analysis in the Environmental Sciences*, 1nd ed. Wiley, 2011.

[96] Chapman, A. P., Jagadish, H. V., and Ramanan, P. Efficient provenance storage. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 2008), SIGMOD '08, ACM, pp. 993–1006.

[97] Chatfield, C. *The Analysis of Time Series: An Introduction*, 6th ed. CRC Press, Florida, US, 2004.

[98] Chen, L., Yang, X., and Tao, F. A semantic web service based approach for augmented provenance. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* (Washington, DC, USA, 2006), WI '06, IEEE Computer Society, pp. 594–600.

[99] Cheney, J., Acar, U. A., and Ahmed, A. Provenance traces. *CoRR abs/0812.0564* (2008).

[100] Cheney, J., Chiticariu, L., and Tan, W. Provenance in databases: Why, how, and where. *Found. Trends databases 1*, 4 (Apr. 2009), 379–474.

[101] Chiticariu, L. Dbnotes: A post-it system for relational databases based on provenance. In *in SIGMOD 05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (2005), ACM Press, pp. 942–944.

[102] Cleveland, W. S. Robust locally weighted regression and smoothing scatterplots. *Journal of thee American Statistical Association 74* (1979).

[103] Cleveland, W. S., and Loader, C. L. *Smoothing by Local Regression: Principles and Methods*. Springer, New York, 1996.

[104] Clifford, B., Foster, I., Voeckler, J., Wilde, M., and Zhao, Y. Tracking provenance in a virtual data grid. *Concurr. Comput.: Pract. Exper. 20*, 5 (Apr. 2008), 565–575.

[105] Compton, M., and et al. The ssn ontology of the w3c semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web 17*, 0 (2012).

[106] Corcho, O., Fernandez-Lopez, M., and Gomez-Perez, A. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & Knowledge Engineering 46*, 1 (2003), 41–64.

[107] Cromwell, J. B., Labys, W. C., and Terraza, M. *Univariate Tests for Time Series Models*. SAGE Publications, 1994.

[108] Cryer, J. D. *Time Series Analysis: With Applications in R*, 2nd ed. ed. Springer texts in statistics. Springer, New York, 2008.

[109] Cui, Y., Widom, J., and Wiener, J. L. Tracing the lineage of view data in a warehousing environment. *ACM Trans. Database Syst. 25*, 2 (June 2000), 179–227.

[110] da Silva, P. P., McGuinness, D. L., and McCool, R. Knowledge provenance infrastructure. *IEEE Data Eng. Bull. 26*, 4 (2003), 26–32.

[111] Da Silva, P. P., Salayandia, L., Del Rio, N., and Gates, A. Q. On the use of abstract workflows to capture scientific process provenance. In *Proceedings of the 2nd conference on Theory and practice of provenance* (Berkeley, CA, USA, 2010), TAPP'10, USENIX Association, pp. 10–10.

[112] de Bruijn, J., Ehrig, M., Feier, C., Martins-Recuerda, F., Scharffe, F., and Weiten, M. *Ontology Mediation, Merging, and Aligning.* July 2006.

[113] de Bruijn, J., Martin-Recuerda, F., Ehrig, M., Polleres, A., and Predoiu, L. D4.4.1 ontology mediation management v1. Tech. Rep. V. 1, Feb. 2005. Digital Enterprise Research Institute, University of Innsbruck.

[114] Dejing, D., Drew, M., and Peishen, Q. Ontology translation on the semantic web. In *Journal of Data Semantics* (2003).

[115] Dielman, T. E. *Applied Regression Analysis. A Second Course in Business and Economic Statistics*, 4ed. ed. South -Western Centage Learning., 2005.

[116] DiGiuseppe, N., Pouchard, L. C., and Noy, N. F. Sweet ontology coverage for earth system sciences. *Springer-Verlag Berling Heidelberg* (2014).

[117] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J. Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management* (New York, NY, USA, 2004), CIKM 04, ACM, pp. 652–659.

[118] DuCharme, B. *Learning SPARQL.* Oreilly and Associate Series. O'Reilly Media, 2011.

[119] Eckner, A. A framework for the analysis of unevenly-spaced time series data. Tech. rep., Dec. 2012.

[120] Ehlers, R. Inferência estatística. Tech. rep., 2009. Departamento de Matemática Aplicada e Estatística Instituto de Ciências Matemáticas e de Computação - USP.

[121] Elsner, J. B., and Tsonis, A. A. *Singular Spectrum Analysis: A New Tool in Time Series Analysis.* Language of science. Springer, 1996.

[122] Erdogan, E., Beygelzimer, A., and Rish, I. Statistical models for unequally spaced time series. *Computer Science* (2005), 626–630.

[123] Euzenat, J., and Shvaiko, P. *Ontology matching.* Springer, Heidelberg (DE), 2007.

[124] Falbo, R. A., Guizzardi, G., and Guizzardi, R. S. S. A importância de ontologias de fundamentação para a engenharia de ontologias de domínio: o caso do domínio de processos de software. *IEEE Transactions Latin America 6* (2008), 244–251.

[125] Feng, G. A note on data smoothing by cubic spline filters. *Signal Processing. IEEE Transactions on 47*, 9 (2002).

[126] Fernandez-Lopez, M., and Gómez-Pérez, A. Overview and analysis of methodologies for building ontologies. *The knowledge Engineering Review 17*, 2 (2002), 129–156.

[127] Fernandez-Lopez, M., Gomez-Perez, A., and Juristo, N. Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium* (Stanford, USA, Mar. 1997), pp. 33–40.

[128] Flandrin, P., Gonçalves, P., and Rilling, G. Detrending and denoising with empirical mode decompositions. In *EUSIPCO-04* (2004), pp. 1581–1584.

[129] Foster, I., Vockler, J., Wilde, M., and Zhao, Y. The virtual data grid: A new model and architecture for data-intensive collaboration. In *In Conference on Innovative Data Systems Research (CIDR), Pacific Grove, CA, USA.* (2003).

[130] Freire, J., Koop, D., Santos, E., and Silva, C. T. Provenance for computational tasks: A survey. *Computing in Science and Engg. 10*, 3 (May 2008), 11–21.

[131] Freitas, A., Knap, T., O'Riain, S., and Curry, E. W3p: Building an opm based provenance model for the web. *Future Generation Computer Systems 27*, 6 (June 2011), 766–774.

[132] Friedman, J. H. Multivariate adaptive regression splines. *The Annals of Statistics 19*, 1 (1991), 1–67.

[133] Gil, Y., Cheney, J., Groth, P., Hartig, O., Miles, S., Moureau, L., and da Silva, P. P. Provenance xg final report. Tech. rep., W3C Incubator Group Report, 2010.

[134] Glavic, B., and Alonso, G. Perm: Processing provenance and data on the same data model through query rewriting. In *In ICDE 09: Proceedings of the 25th International Conference on Data Engineering* (2009), pp. 174–185.

[135] Glavic, B., and Dittrich, K. R. Data provenance: A categorization of existing approaches. In *Datenbanksysteme in Business, Technologie und Web (BTW 2007)* (2007), pp. 227–241.

[136] Golyandina, N., Nekrutkin, V., and Zhigljavsky, A. A. *Analysis of Time Series Structure: SSA and Related Techniques.* Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 2001.

[137] Gotz, D., and Zhou, M. X. Characterizing users' visual analytic activity for insight provenance. *Information Visualization 8*, 1 (2009), 42–55.

[138] Green, T. J., Karvounarakis, G., and Tannen, V. Provenance semirings. In *PODS '07: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (New York, NY, USA, 2007), ACM, pp. 31–40.

[139] Greenwood, M., Goble, C., Stevens, R., Zhao, J., Addis, M., Marvin, D., Moreau, L., and Oinn, T. Provenance of e-Science experiments-experience from bioinformatics. *Proceedings of the UK OST e-Science second All Hands Meeting 4* (2003).

[140] Groth, D. P., and Streefkerk, K. Provenance and annotation for visual exploration systems. *IEEE Trans. Vis. Comput. Graph. 12*, 6 (2006), 1500–1510.

[141] Gruber, T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition 5*, 2 (1993), 199–220.

[142] Gruninger, M., and Fox, M. S. Methodology for the design and evaluation of ontologies. In *International Joint Conference on Artificial Inteligence (IJCAI95), Workshop on Basic Ontological Issues in Knowledge Sharing* (1995).

[143] Guarino, N. Formal ontology and information systems. In *Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998* (Amsterdam, 1998), N. Guarino, Ed., IOS Press, pp. 3–15.

[144] Hair, J. F., Tatham, R. L., Anderson, R. E., and Black, W. *Multivariate Data Analysis (5th Edition)*, 5th ed. Prentice Hall, Mar. 1998.

[145] Hair, J. F., Tatham, R. L., Anderson, R. E., and Black, W. *Multivariate Data Analysis (7th Edition)*, 7th ed. Pearson Prentice Hall, 2010.

[146] Hanisch, R. J., Farris, A., Greisen, E. W., Pence, W. D., Schlesinger, B. M., Teuben, P. J., Thompson, R. W., and Warnock III, A. Definition of the flexible image transport system (fits). *A&A 376* (2001), 359–380.

[147] Hansen, L. P. Large sample properties of generalized method of moments estimators. *Econometrica 50*, 4 (1982), 1029–1054.

[148] Hartig, O. Provenance information in the web of data. *LDOW 2009* (Apr. 2009).

[149] Hebeler, J., Fisher, M., Blace, R., Perez-Lopez, A., and Dean, M. *Semantic Web Programming*. John Wiley and Sons Inc., Chichester, West Sussex, Hoboken, NJ, 2009.

[150] Henson, C. A., H. Neuhaus, H., Sheth, A. P., Thirunarayan, K., and Buyya, R. An ontological representation of time series observations on the semantic sensor web. *Proceedings of 1st International Workshop on the Semantic Sensor Web* (2009).

[151] Hitzler, P., Krtzsch, M., and Rudolph, S. Semantic web modelling languages (part 2) tutorial. In *IJCAI* (2009), p. 66 pp.

[152] Horridge, M., Jupp, S., Moulton, G., Rector, A. R., and Stevens, R.and Wroe, C. A practical guide to building owl ontologies using protege 4 and co-ode tools. edition 1.1.

[153] Horrocks, I., Kutz, O., and Sattler, U. The even more irresistible sroiq. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006)* (2006), pp. 57–67.

[154] Horrocks, I., Schneider, P. F. P., Bechhofer, S., and Tsarkov, D. Owl rules: A proposal and prototype implementation. *Journal of Web Semantics 3* (2005), 23–40.

[155] Hosking, J. R. M. L-moments: Analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society. Series B (Methodological) 52*, 1 (1990), 105–124.

[156] Huang, N. E., and et al. The empirical mode decomposition and the hubert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A 454*, 1971 (1998).

[157] Islam, S. Provenance, lineage, and workflows. Tech. rep., May 2010. Computer Science Department, Brown University, RI, USA Advised by Stanley B. Zdonik.

[158] Isnanto, R. R. Comparison on several smoothing methods in nonparametric regression. *Journal System Computer 1*, 1 (2011).

[159] Jackman, S. Maximum likelihood. Tech. rep., 2006. Department of Political Science, Stanford University.

[160] Jagadish, H. V., and Olken, F. Database management for life sciences research. *SIGMOD Rec. 33*, 2 (June 2004), 15–20.

[161] Jia, S., Guo, Y., Wang, Q., and Zhang, J. Trend extraction and similarity matching of financial time series based on emd method. In *CSIE (4)* (2009), M. Burgin, M. H. Chowdhury, C. H. Ham, S. A. Ludwig, W. Su, and S. Yenduri, Eds., IEEE Computer Society, pp. 526–530.

[162] Kalyanpur, A., Parsia, B., Sirin, E., Grau, B. C., and Hendler, J. Swoop: A web ontology editing browser. *Journal of Web Semantics 4* (2005).

[163] Kim, J., Deelman, E., Gil, Y., Mehta, G., and Ratnakar, V. Provenance trails in the wings-pegasus system. *Concurr. Comput.:Pract. Exper. 20*, 5 (Apr. 2008), 587–597.

[164] Kim, S. J., Koh, K., Boyd, S., and Gorinevsky, D. l1 trend filtering. *Siam Review. Society for Industrial and Applied Mathematics 51*, 2 (2009), 339–360.

[165] Kiryakov, A. Ontologies for knowledge management. *DAVIES, J. and et al. (Eds). Semantic Web Technologies: Trends and Research in Ontology-Based Systems* (2006), 115–138.

[166] Knublauch, H., Fergerson, R. W., Noy, N. F., and Musen, M. A. The protege owl plugin: An open development environment for semantic web applications. In *Semantic Web - ISWC 2004, Proceedings* (2004), vol. 3298, pp. 229–243. 3rd International Semantic Web Conference, Hiroshima, JAPAN, NOV 07-11, 2004.

[167] Kovacs, G., and Bakos, G. A. Application of the trend filtering algorithm in the search for multiperiodic signals. *Communications in Asteroseismology 157* (Dec. 2008), 82–86.

[168] Lando, P., Lapujade, A., Kassel, G., and Furst, F. An ontological investigation in the field of computer programs. *In Software and Data Technologies, Communications in Computer and Information Science. 22* (2009), 371–383.

[169] Lange, C. Ontologies and languages for representing mathematical knowledge on the semantic web. *Semantic Web 4*, 2 (2013), 119–158.

[170] Lassila, O., and McGuinness, D. L. The role of frame-based representation on the semantic web. Tech. Rep. KSL-01-02, Stanford University, Stanford, 2001.

[171] Leymann, F. Web services flow language (wsfl 1.0). `URL:http://www.ibm.com/software/solutions/webservices/pdf/WSFL.pdf`, May 2013.

[172] Li, L., Li, K., Liu, C., and Liu, C. Comparison of detrending methods in spectral analysis of heart rate variability. *Research Journal of Applied Sciences, Engineering and Technology 3*, 9 (2011), 1014–1021.

[173] Li, X., Lebo, T., and McGuinness, D. L. Provenance-based strategies to develop trust in semantic web applications. In *IPAW* (2010), D. L. McGuinness, J. Michaelis, and L. Moreau, Eds., vol. 6378 of *Lecture Notes in Computer Science*, Springer, pp. 182–197.

[174] Llaves, A., and Renschler, C. S. Observing changes in real-time sensor observations. In *Multidisciplinary Research on Geographical Information in Europe and Beyond. Proceedings of the AGILE'2012 International Conference on Geographic Information Science.*

[175] Loader, C. Smoothing: Local regression techniques. *Center for Applied Statistics and Economics (CASE) 12* (2004).

[176] Malaverri, J. G., Medeiros, C. B., and Lamparelli, R. C. A provenance approach to assess the quality of geospatial data. In *SAC* (2012), S. Ossowski and P. Lecca, Eds., ACM, pp. 2043–2044.

[177] Mandel, J. Use of the singular value decomposition in regression analysis. *The American Statistician 36*, 1 (1982).

[178] Marathe, A. P. Tracing lineage of array data. In *Proceedings of the 13th International Conference on Scientific and Statistical Database Management* (Washington, DC, USA, 2001), SSDBM '01, IEEE Computer Society.

[179] Marcotte, D. Linear regression. Tech. rep., 2009. B. Govaerts - Institut de Statistique - UCL.

[180] Mazeh, T., and et al. Removing systematics from the CoRoT light curves: I. Magnitude-Dependent Zero Point. *arXiv:0907.2237v1* (2009).

[181] Mazeh, T., Tamuz, O., and Zucker, S. The sys-rem detrending algorithm: Implementation and testing. *astro-ph/0612418* (Dec. 2006).

[182] McGuinness, D. L., Ding, L., da Silva, P. P., and Chang, C. Pml 2: A modular explanation interlingua. In *In Proceedings of the AAAI 2007 Workshop on Explanation-aware Computing* (2007), pp. 22–23.

[183] Meinl, T. *A Novel Wavelet Based Approach for Time Series Data Analysis*. PhD thesis, Karlsruhe, 2011.

[184] Meko, D. Geos 585a applied time series analysis. Tech. rep., 2011. Time Series Course.

[185] Michaelis, J. L., and et al. Towards usable and interoperable workflow provenance: Empirical case studies using pml. In *Proceedings of the First International Workshop on the Role of Semantic Web in Provenance Management* (2009), pp. TW–2009–20.

[186] Michaelis, J, R., Zednik, S., Ding, L., and McGuinness, D. L. A comparison of the opm and pml provenance models. In *Tetherless World Constellation (RPI) Technical Report* (2009).

[187] Michlmayr, A., Rosenberg, F., Leitner, P., and Dustdar, S. Service provenance in qos-aware web service runtimes. In *Proceedings of the 2009 IEEE International Conference on Web Services* (Washington, DC, USA, 2009), ICWS '09, IEEE Computer Society, pp. 115–122.

[188] Miles, S., Deelman, E., Groth, P., Vahi, K., Mehta, G., and Moreau, L. Connecting scientific data to scientific experiments with provenance. In *Proceedings of the Third IEEE International Conference on e-Science and Grid Computing* (Washington, DC, USA, 2007), E-SCIENCE '07, IEEE Computer Society, pp. 179–186.

[189] Mislis, D., Hodgkin, S., and Birkby, J. Detrend survey transiting light curves algorithm (dstl). *Astron. Soc.* (2002).

[190] Mislis, D., Schmitt, J. H. M. M., Carone, L., Guenther, E. W., and Patzold, M. An algorithm for correcting corot raw light curves. *arXiv/1008.0300* (2010).

[191] Misra, A., Blount, M., Kementsietsidis, A., Sow, D., and Wang, M. Provenance and annotation of data and processes. Springer-Verlag, Berlin, Heidelberg, 2008, ch. Advances and Challenges for Scalable Provenance in Stream Processing Systems, pp. 253–265.

[192] Moghtaderi, A., Flandrin, P., and Borgnat, P. Trend filtering via empirical mode decompositions. *Computational Statistics and Data Analysis 58* (2013), 114–126.

[193] Montesino-Pouzols, F., and Lendasse, A. Effect of different detrending approaches on computational intelligence models of time series. In *IJCNN* (2010), IEEE, pp. 1–8.

[194] Moreau, L. The foundations for provenance on the web. *Foundations and Trends in Web Science 2*, 2-3 (2010), 99–241.

[195] Moreau, L., and et al. Special issue: The first provenance challenge. *Concurrency and Computation: Practice and Experience 20*, 5 (2008), 409–418.

[196] Moreau, L., and et al. The open provenance model core specification (v1.1). *Future Gener. Comput. Syst. 27*, 6 (June 2011), 743–756.

[197] Moreau, L., Freire, J., Futrelle, J., Mcgrath, R., Myers, J., and Paulson, P. The open provenance model: An overview. 2008, pp. 323–326.

[198] Morettin, P. A., and Toloi, C. M. C. *Análise de Séries Temporais*. São Paulo: Blucher, 2006.

[199] Myers, J. D., Chappell, A., Elder, M., Geist, A., and Schwidder, J. Re-integrating the research record. *Computing in Science and Engg. 5*, 3 (May 2003), 44–50.

[200] Myers, J. D., and et al. Embedding data within knowledge spaces. *CoRR abs/0902.0744* (2009).

[201] Nadaraya, E. A. On estimating regression. *Theory of Probability and Its Applications 9* (1964).

[202] Nagpaul, P. S. Time series analysis in winidams. Tech. rep., 2005.

[203] Natrella, M. *Nist. Sematech. e-Handbook of Statistical Methods*. NIST. SEMATECH, July 2010.

[204] Noy, N. F., and McGuinness, D. L. Ontology development 101: A guide to creating your first ontology. *Development 32*, 1 (2001), 1–25.

[205] Noy, N. F., and Musen, M. A. Smart:automated support for ontology merging and alignment. In *Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management (KAW'99)* (Alberta, Oct. 1999).

[206] Ofir, A., and et al. The sars algorithm: Detrending corot light curves with sysrem using simultaneous external parameters. *arXiv/1003.0427* (2010).

[207] Pena, D., Tiao, G. C., and Tsay, R. S. *A Course in Time Series Analysis*. John Wiley and Sons, Inc., 2001.

[208] Pinto, H. S., and Martins, J. P. Ontologies: How can they be built? *Knowledge and Information Systems 6*, 4 (July 2004), 441–464.

[209] Plale, B., Alameda, J., Wilhelmson, B., Gannon, D., Hampton, S., Rossi, A., and Droegemeier, K. Active management of scientific data. *IEEE Internet Computing 9*, 1 (Jan. 2005), 27–34.

[210] Pollock, D. S. G. *A Handbook of Time series Analysis, Signal Processing and Dynamics*. No. 1 in A Handbook of Time series Analysis, Signal Processing and Dynamics. Academic, 1999.

[211] Pollock, D. S. G. Trend estimation and de-trending via rational square-wave filters. *Journal of Econometrics 99*, 2 (2000).

[212] Poularikas, A. D. *The Handbook of Formulas and Tables of Signal Processing*. CRC Press LLC, 1999.

[213] Prud'hommeaux, E., and Seaborne, A. Sparql query language for rdf. `URL: http://www.w3.org/TR/rdf-sparql-query/`, Jan. 2014.

[214] Racine, J. Cran.r project. `URL:http://cran.r-project.org/web/packages/crs/vignettes/spline_primer.pdf`, Aug. 2013.

[215] Ram, S., and Liu, J. A new perspective on semantics of data provenance. In *SWPM* (2009).

[216] Raskin, R. G. Sweet 2.1 ontologies. *AGU Fall Meeting Abstracts* (2010).

[217] Re, C., and Suciu, D. Approximate lineage for probabilistic databases. *Proc. VLDB Endow. 1*, 1 (Aug. 2008), 797–808.

[218] Rehfeld, K., Marwan, N., Heitzig, J., and Kurths, J. Comparison of correlation analysis techniques for irregularly sampled time series. *Nonlinear Processes in Geophysics 18*, 3 (June 2011), 389–404.

[219] Sahoo, S. S., and et al. Ontology-driven provenance management in escience: An application in parasite research. In *Proceedings of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part II* (Berlin, Heidelberg, 2009), OTM '09, Springer-Verlag, pp. 992–1009.

[220] Sahoo, S. S., Sheth, A., and Henson, C. Semantic provenance for escience. In *IEEE Internet Computing* (2008), pp. 46–54.

[221] Scheidegger, C., Koop, D., Santos, E., Vo, H., Callahan, S., Freire, J., and Silva, C. Tackling the provenance challenge one layer at a time. *Concurr. Comput.:Pract. Exper. 20*, 5 (Apr. 2008), 473–483.

[222] Sheth, A., Henson, C., and Sahoo, S. S. Semantic sensor web. *IEEE Internet Computing 12*, 4 (2008), 78–83.

[223] Shumway, R. H., and Stoffer, D. S. *Time Series Analysis and Its Applications: With R Examples*, 2nd ed. Springer, May 2006.

[224] Silberschatz, A., Korth, H., and Sudarshan, S. *Database Systems Concepts*, 5 ed. McGraw-Hill, Inc., New York, NY, USA, 2006.

[225] Simmhan, Y. L. *Provenance Framework in Support of Data Quality Estimation*. PhD thesis, Indianapolis, IN, USA, 2007. AAI3297094.

[226] Simmhan, Y. L., Plale, B., and Gannon, D. A survey of data provenance techniques. Tech. rep., Computer Science Departament, Indiana University, Bloomington IN 47405, 2005.

[227] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. Pellet: A practical owl-dl reasoner. *Web Semant. 5*, 2 (June 2007), 51–53.

[228] Smith, S. W. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[229] Souilah, I., Francalanza, A., and Sassone, V. A formal model of provenance in distributed systems. In *First workshop on on Theory and practice of provenance* (Berkeley, CA, USA, 2009), TAPP'09, USENIX Association, pp. 1:1–1:11.

[230] Spiegel, M. R. *Estatística*. McGraw-Hill, São Paulo, 1985.

[231] Staab, S., Studer, R., Schnurr, H., and Sure, Y. Knowledge processes and ontologies. *IEEE Intelligent Systems 16*, 1 (2001), 26–34.

[232] Stadnytska, T. Deterministic or stochastic trend. decision on the basis of the augmented dickey-fuller test. *Computational Statistics and Data Analysis 6*, 2 (2010), 83–92.

[233] SuarezFigueroa, M. C., Gomez-Perez, A., Motta, E., and Gangemi, A. *Ontology Engineering in a Networked World*. Springer-Verlag Berlin Heidelberg, 2012.

[234] Tan, W. C. Provenance in databases: Past, current, and future. *IEEE Data Eng. Bull. 30*, 4 (2007), 3–12.

[235] Tautz, C., and von Wangenheim, C. *REFSENO: A Representation Formalism for Software Engineering Ontologies*. IESE-Report Fraunhofer Einrichtung experimentelles Software Engineering. Fraunhofer-IESE, 1998.

[236] Uschold, M., and Gruninger, M. Ontologies: Principles, methods and applications. *Knowledge Engineering Review 11*, 2 (June 1996), 93–155.

[237] Uschold, M., and King, M. Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95* (Montreal, Canada, 1995).

[238] van Heijst, G., Schreiber, A. T., and Wielinga, B. J. Using explicit ontologies in kbs development. *International Journal Human-Computer Studies 46*, 2 (1997), 183–292.

[239] Vautard, R. P. Y., and Ghil, M. Singular spectrum analysis in nonlinear dynamics with applications to paleoclimatic time series. *Physica D 35* (1989), 395–424.

[240] Wahba, G. Spline functions for observational data. *SIAM, Philadelphia* (1990).

[241] Wand, M. P., and Jones, M. C. Kernel smoothing. *Chapman and Hall, London,* (1995).

[242] Wang, Y. R., and Madnick, S. E. A polygen model for heterogeneous database systems: The source tagging perspective. In *Proceedings of the 16th International Conference on Very Large Data Bases* (San Francisco, CA, USA, 1990), VLDB '90, Morgan Kaufmann Publishers Inc., pp. 519–538.

[243] Wedderburn, R. W. M. Quasi-likelihood functions, generalized linear models, and the gauss-newton method. *Biometrika 61*, 3 (1974).

[244] Wei, W. S. W. *Time Series Analysis. Univariate and Multivariate Methods*, 2nd ed. Pearson Education, 2006.

[245] Widom, J. Trio: A system for integrated management of data, accuracy, and lineage. In *Proc. of the 2nd Biennial Conference on Innovative Data Systems Research (CIDR)* (Jan. 2005).

[246] Woodruff, A., and Stonebraker, M. Supporting fine-grained data lineage in a database visualization environment. In *Proceedings of the Thirteenth International Conference on Data Engineering* (Washington, DC, USA, 1997), ICDE '97, IEEE Computer Society, pp. 91–102.

[247] Wu, Z., and Huang, N. E. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in Adaptive Data Analysis 1*, 1 (2009), 1–41.

[248] Wu, Z., Huang, N. E., Long, S. R., and Peng, C. On the trend, detrending, and variability of nonlinear and nonstationary time series. *Proceedings of the National Academy of Sciences 104*, 38 (2007), 14889–14894.

[249] Yaffee, R. A., and McGee, M. *Introduction to Time Series Analysis and Forecasting: With Applications of SAS and SPSS*, 1st ed. Academic Press, Inc., Orlando, FL, USA, 2000.

[250] Yu, L. *Introduction to the Semantic Web and Semantic Web Services*, 1 ed. Chapman and Hall CRC, 2007.

[251] Zachary, I., Nitin, K., Aneesh, K., and Murat, C. Orchestra: Rapid, collaborative sharing of dynamic data. In *In CIDR* (2005).

[252] Zhang, Y., Vasconcelos, W., and Sleeman, D. Ontosearch: An ontology search engine. In *Research and Development in Intelligent Systems XXI* (2005), M. Bramer, F. Coenen, and T. Allen, Eds., Springer London, pp. 58–69.

[253] Zhao, J., and et al. Annotating, linking and browsing provenance logs for e-science. In *In Proc. of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data* (2003), pp. 158–176.

[254] Zhao, J., Goble, C., Stevens, R., and Turi, D. Mining taverna's semantic web of provenance. *Concurr. Comput.: Pract. Exper. 20*, 5 (Apr. 2008), 463–472.

[255] Zhao, J., Goble, C. A., Stevens, R., and Bechhofer, S. Semantically linking and browsing provenance logs for e-science. In *ICSNW* (2004), vol. 3226 of *Lecture Notes in Computer Science*, Springer, pp. 158–176.

APÊNDICE A

TERMO DE COMPROMISSO E PARECER DO AVALIADOR NA ONTOLOGIA TSO

O presente documento visa esclarecer ao avaliador da ontologia questões relacionadas ao desenvolvimento e ao processo de avaliação da Ontologia *Time Series Ontology*.

1. Propósito da Ontologia

A ontologia proposta é intitulada *Time Series Ontology*, a qual tem por objetivo enriquecer semanticamente, por meio da proveniência de dados, sequências de observações regulares e ordenadas no tempo (séries temporais).

2. Conteúdo da Ontologia

A ontologia apresenta a definição dos principais conceitos e relacionamentos relacionados às séries temporais, incluindo suposições, modelos e tipos de decomposição, componentes e características dos componentes.

3. Escopo da Ontologia

O escopo da ontologia relaciona-se com séries temporais não-estacionárias, ou seja, que apresentam algum tipo de tendência. Destaca-se que na ontologia em questão não são incluídos métodos estatísticos para transformação das séries.

4. Processo de Avaliação

O processo de avaliação tem por objetivo contribuir com o desenvolvimento da ontologia, envolvendo a avaliação da definição de conceitos e relacionamentos por uma comunidade, com o objetivo de utilização por agentes humanos e/ou máquinas, assim como permitir reuso e/ou expansão da mesma. Os resultados da avaliação contribuem para a definição de um vocabulário controlado nesse contexto, permitindo enriquecer semanticamente as séries temporais.

5. Avaliadores e Questões éticas

O processo de avaliação inclui participantes das áreas de Análise de Séries Temporais, assim como desenvolvedores de Ontologias. A avaliação ocorre de modo informal, ou seja, não serão divulgados nomes dos avaliadores, apenas contribuições. Esteja à vontade para sugerir, criticar, enfim, propor alterações que visem contribuir para o desenvolvimento da presente ontologia. Desde já agradecemos sua participação.

PARECER DO AVALIADOR DA ONTOLOGIA

Ontologia Avaliada: *Time Series Ontology* Prefixo: *tso*

Data:__/__/____ Avaliador:

1) Você concorda com o levantamento das questões de competência que a ontologia deve ser capaz de responder?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestão:

2) Você está de acordo com a definição das classes e subclasses da ontologia?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

3) Você está de acordo com a definição dos relacionamentos da ontologia?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

4) Você está de acordo com as instâncias da ontologia?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

5) O vocabulário proposto está em conformidade com termos utilizados em Análise de Séries Temporais?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestão:

6) Você está de acordo com as bibliografias utilizadas para definição dos termos propostos?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestão:

7) Na sua opinião, a ontologia em questão pode vir a ser utilizada como forma de adicionar conhecimento semântico às séries temporais?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

8) Espaço aberto para comentários, críticas, sugestões de ampliação de conteúdo, propostas de alteração de nomenclatura, entre outras contribuições.

APÊNDICE B

TERMO DE COMPROMISSO E PARECER DO AVALIADOR NA ONTOLOGIA DO

O presente documento visa esclarecer ao avaliador da ontologia questões relacionadas ao desenvolvimento e ao processo de avaliação da Ontologia *Detrend Ontology*.

1. Propósito da Ontologia

A ontologia proposta é intitulada *Detrend Ontology*, a qual tem por objetivo enriquecer semanticamente o passo de extração de tendências da fase de pré-processamento da análise das séries temporais.

2. Conteúdo da Ontologia

A ontologia em questão apresenta definições, relacionamentos e instâncias dos principais métodos que podem ser usados para *detrending*, incluindo a estimação de tendências por meio da análise de regressão paramétrica e não-paramétrica e o uso de filtros.

3. Escopo da Ontologia

O escopo da ontologia relaciona-se a métodos de transformação de séries temporais não-estacionárias no domínio do tempo, podendo ser estendida para o domínio da frequência. Destaca-se que demais métodos da fase de pré-processamento das séries temporais, tais como detecção e remoção de *outliers*, *clustering*, entre outros, não são incluídos na ontologia em questão.

4. Processo de Avaliação

O processo de avaliação tem por objetivo contribuir com o desenvolvimento da ontologia, envolvendo a avaliação da definição de conceitos e relacionamentos por uma comunidade, com o objetivo de utilização por agentes humanos e/ou máquinas, assim como permitir reuso e/ou expansão da mesma. Os resultados da avaliação contribuem para a definição de um vocabulário controlado nesse contexto, permitindo enriquecer semanticamente o passo de *detrending* da análise de séries temporais.

5. Avaliadores e Questões éticas

O processo de avaliação inclui participantes das áreas de Análise de Séries Temporais, assim como desenvolvedores de Ontologias. A avaliação ocorre de modo informal, ou seja, não serão divulgados nomes dos avaliadores, apenas contribuições. Esteja à vontade para sugerir, criticar, enfim, propor alterações que visem contribuir para o desenvolvimento da presente ontologia. Desde já agradecemos sua participação.

PARECER DO AVALIADOR DA ONTOLOGIA

Ontologia Avaliada: *Detrend Ontology* Prefixo: *do*

Data:__/__/____ Avaliador:

1) Você concorda com o levantamento das questões de competência que a ontologia deve ser capaz de responder?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestões:

2) Você está de acordo com o reuso e extensão da Ontologia StatisticalAnalysis (prefixo a:) (URL:http://a.com/StatisticalAnalysis#) para modelagem da Análise de Regressão?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestão de reuso de outra(s) ontologia(s) de Análise de Regressão:

3) Você concorda com o reuso de conceitos, relações e instâncias a partir de *Semantic Web for Earth and Environmental Terminology - SWEET Ontologies* (http://sweet.jpl.nasa.gov/2.3/sweetAll.owl)?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestão de reuso de termos a partir de outra(s) ontologia(s):

4) Você está de acordo com a definição das classes e subclasses da ontologia? Ex.: do:Domain
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

5) Você está de acordo com a definição dos relacionamentos da ontologia? Ex.: do:hasDomain
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

6) Você está de acordo com as instâncias (indivíduos) da ontologia? Ex.:do:Moving_Average
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

7) O vocabulário proposto está em conformidade com a nomenclatura utilizada em métodos estatísticos aplicados no pré-processamento de séries temporais?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestão:

8) Você está de acordo com as bibliografias utilizadas para definição dos conceitos propostos?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestão de bibliografia(s):

9) Na sua opinião, a ontologia em questão pode vir a ser utilizada como forma de adicionar conhecimento semântico quanto aos métodos estatísticos relacionados ao passo

de extração de tendências (*detrending*)?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

10) Espaço aberto para comentários, críticas, sugestões de ampliação de conteúdo, propostas de alteração de nomenclatura, entre outras contribuições.

Assinatura do Avaliador

APÊNDICE C

TERMO DE COMPROMISSO E PARECER DO AVALIADOR NA ONTOLOGIA DPM

O presente documento visa esclarecer ao avaliador da ontologia questões relacionadas ao desenvolvimento e ao processo de avaliação da Ontologia *Detrend Provenance Model*.

1. Propósito do Modelo de Proveniência para Extração de Tendências em Séries Temporais

O presente modelo de proveniência relaciona-se com o desenvolvimento de uma ontologia em linguagem *Ontology Web Language* - OWL. O modelo proposto é intitulado *Detrend Provenance Model* (prefixo dpm), apresentando como objetivo principal enriquecer semanticamente o passo de extração de tendências da fase de pré-processamento da análise de séries temporais, por meio da geração e inferência de informações de proveniência, permitindo interoperabilidade semântica na correção das séries temporais.

2. Conteúdo da Ontologia

A ontologia em questão é desenvolvida de forma modular, incluindo os seguintes módulos (ontologias):

- *Open Provenance Model* - OPM (namespace opmo:)
- *Time Series Ontology* - TSO (namespace tso:)
- *Detrend Ontology* - DO (namespace do:)

Destaca-se que tais módulos também reutilizam conceitos e/ou importam outras ontologias.

3. Escopo da Ontologia

O escopo do modelo de proveniência relaciona-se a métodos no domínio do tempo para transformação das séries temporais não-estacionárias, ou seja, que apresentam tendências, podendo ser expandido para o domínio da frequência.

4. Processo de Avaliação

O processo de avaliação tem por objetivo contribuir com o desenvolvimento da ontologia, envolvendo a avaliação da definição de conceitos e relacionamentos por uma comunidade, com o objetivo de utilização por agentes humanos e/ou máquinas, assim como permitir reuso e/ou expansão da mesma. Os resultados da avaliação contribuem para a definição de um vocabulário controlado nesse contexto, permitindo enriquecer semanticamente a correção de séries temporais.

5. Avaliadores e Questões éticas

O processo de avaliação inclui participantes das áreas de Análise de Séries Temporais, assim como desenvolvedores de Ontologias. A avaliação ocorre de modo informal, ou seja, não serão divulgados nomes dos avaliadores, apenas contribuições. Esteja à vontade para sugerir, criticar, enfim, propor alterações que visem contribuir para o desenvolvimento da presente ontologia. Desde já agradecemos sua participação.

PARECER DO AVALIADOR DA ONTOLOGIA

Ontologia Avaliada: *Detrend Provenance Model* Prefixo: *dpm*

Data:__/__/____ Avaliador:

1) Você concorda com o levantamento das questões de competência que a ontologia deve ser capaz de responder?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestões:

2) Você está de acordo com o reuso/extensão das seguintes ontologias (módulos):
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestão de reuso de outra(s) ontologia(s):

- *Open Provenance Model* (prefixo opmo:)
  URL: http://openprovenance.org/model/opmo#

- *Time Series Ontology*(prefixo tso:)
  URL: http://www.semanticweb.org/ontologies/2013/7/TimeSeriesOntology.owl#

- *Detrend Ontology* (prefixo do:)
  URL: http://www.semanticweb.org/ontologies/2013/7/DetrendOntology.owl#

3) Você está de acordo com a definição das classes e subclasses do modelo de proveniência proposto?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

4) Você está de acordo com a definição dos relacionamentos do modelo de proveniência proposto?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

5) Você está de acordo com as instâncias do modelo de proveniência proposto?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

6) O vocabulário (proposto ou reutilizado e estendido) está em conformidade com a nomenclatura utilizada em métodos estatísticos aplicados no pré-processamento de séries temporais para que possa gerar proveniência em métodos de extração de tendências?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Sugestão:

7) Na sua opinião, o modelo de proveniência proposto pode vir a ser utilizado como forma de adicionar conhecimento semântico quanto a séries temporais e aos métodos

estatísticos relacionados ao processo de *detrending* da fase de pré-processamento da análise de séries temporais?
( ) Sim ( ) Não ( ) Concordo parcialmente
( ) Parecer:

8) Espaço aberto para comentários, críticas, sugestões de ampliação de conteúdo, propostas de alteração de nomenclatura, entre outras contribuições.

APÊNDICE D

PERFIL DO AVALIADOR DA ONTOLOGIA

1) Qual sua área de atuação?
( ) Economia ( ) Matemática
( ) Estatística ( ) Computação
( ) Física/Astrofísica ( ) Outra:

2) Qual sua experiência com análise de séries temporais?
( ) formação acadêmica ( ) ministro ou já ministrei disciplina
( ) desenvolvimento de projeto de pesquisa ( ) fiz disciplina na área
( ) não tenho experiência nessa área ( ) outra:

3) Qual sua experiência com métodos estatísticos?
( ) formação acadêmica ( ) ministro ou já ministrei disciplina
( ) desenvolvimento de projeto de pesquisa ( ) fiz disciplina na área
( ) não tenho experiência nessa área ( ) outra:

4) Qual sua experiência com desenvolvimento de ontologias?
( ) formação acadêmica ( ) ministro ou já ministrei disciplina
( ) desenvolvimento de projeto de pesquisa ( ) fiz disciplina na área
( ) não tenho experiência nessa área ( ) outra:

5) Qual parte da análise de séries temporais tem experiência (se aplicável):
( ) Modelagem de séries temporais ( ) Pré-processamento de séries temporais
( ) Análise de séries temporais ( ) Previsão de séries temporais
( ) Outra:

6) Há quanto tempo trabalha ou pesquisa com séries temporais e métodos estatísticos (se aplicável):
( ) De 1 a 3 anos ( ) Aproximadamente 5 anos ( ) Mais de 5 anos

7) Há quanto tempo trabalha ou pesquisa com desenvolvimento de ontologias (se aplicável):
( ) De 1 a 3 anos ( ) Aproximadamente 5 anos ( ) Mais de 5 anos

APÊNDICE E

DIAGRAMAS DOS MÉTODOS DE *DETRENDING* E CONSULTAS NA ONTOLOGIA DO

O desenvolvimento da ontologia DO é centrado no reuso, como forma de interoperabilidade. A ontologia reutilizada e estendida é StatisticalAnalysis.owl. Neste apêndice, são apresentados diagramas e consultas. As Figuras E.1 a E.29 apresentam os diagramas dos métodos de *detrending*, como análise de regressão, suavização, filtro e seus parâmetros. As Figuras E.30 a E.52 apresentam consultas relacionadas aos métodos.

Na Figura E.1, a partir da classe (do:ParametricRegressionAnalysis) é feita a associação, usando a propriedade de objeto (do:hasAnalysis), ao respectivo tipo de análise de regressão, a partir do reuso da ontologia StatisticalAnalysis. O relacionamento (do:hasAnalysis) a partir da classe (do:ParametricRegressionAnalysis) pode ser declarada para alguma classe de regressão bivariada (a:LinearRegression) ou (a:NonlinearRegression) ou de regressão múltipla (a:MultipleLinearRegression) ou (do:MultipleNonlinearRegression).

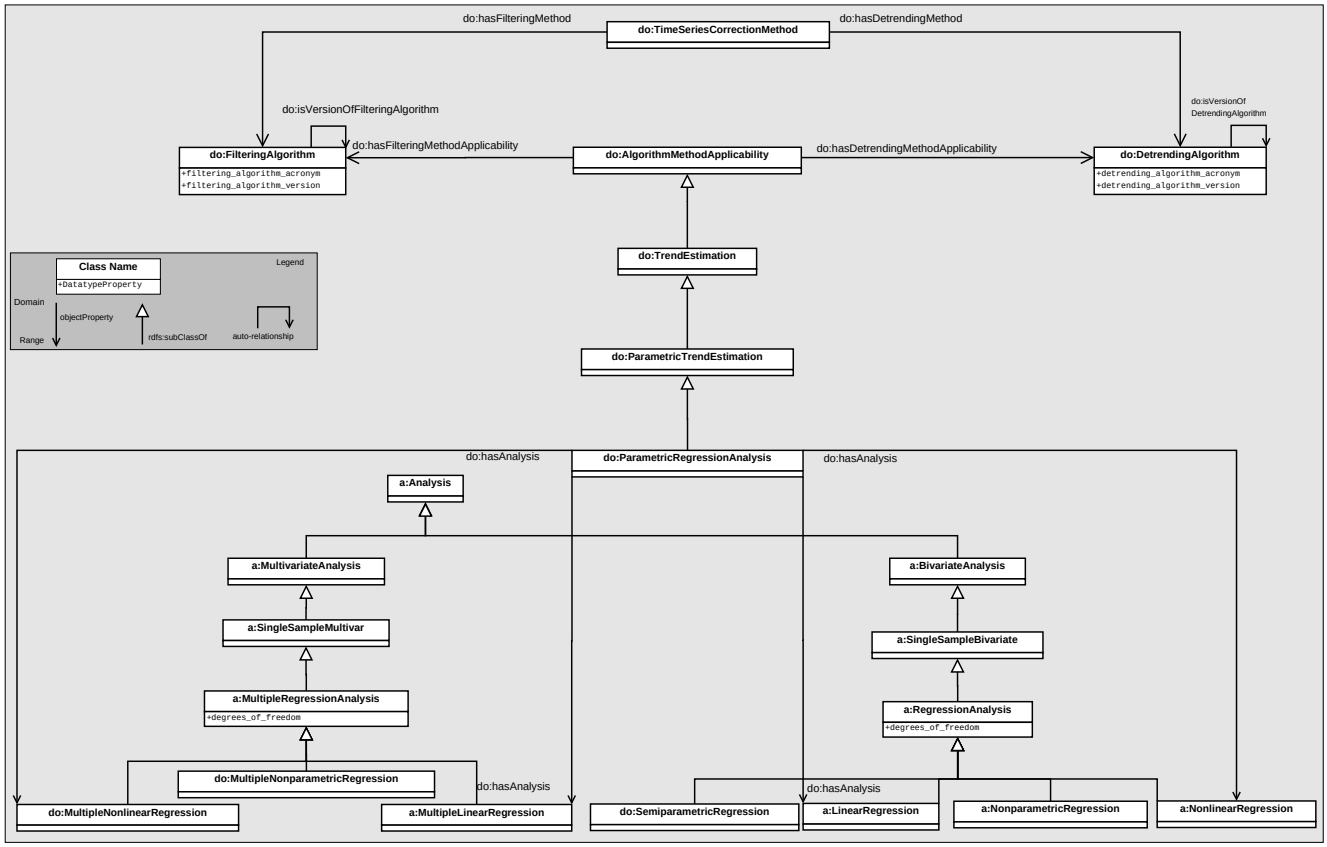


Figura E.1: Classe (do:ParametricTrendEstimation).

A partir do reuso da ontologia StatisticalAnalysis, são estendidos métodos de regressão linear que podem ser usados para *detrending* (Figura E.2), tais como regressão linear mínimos quadrados ordinários, regressão linear ponderada, regressão linear polinomial e regressão linear por partes ou segmentada, onde a regressão é aplicada em segmentos da variável independente. A regressão polinomial é considerada como linear, devido à forma de como os parâmetros entram no modelo, nesse caso, de forma linear. A partir de (a:NonlinearRegression), são criadas as classes de regressão não-linear ponderada e regressão não-linear mínimos quadrados ordinários.

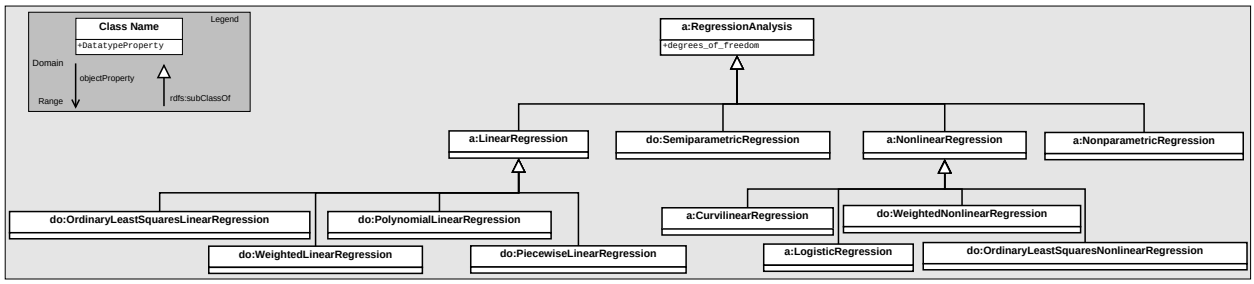


Figura E.2: Classe (a:RegressionAnalysis).

A classe (a:MultipleRegressionAnalysis) (Figura E.3) está relacionada por meio da propriedade (do:hasVariableSelectionMethod) com a classe contendo os métodos de seleção de variáveis independentes, tais como adição *forward*, eliminação *stepwise* e eliminação *backward*, conforme [144].

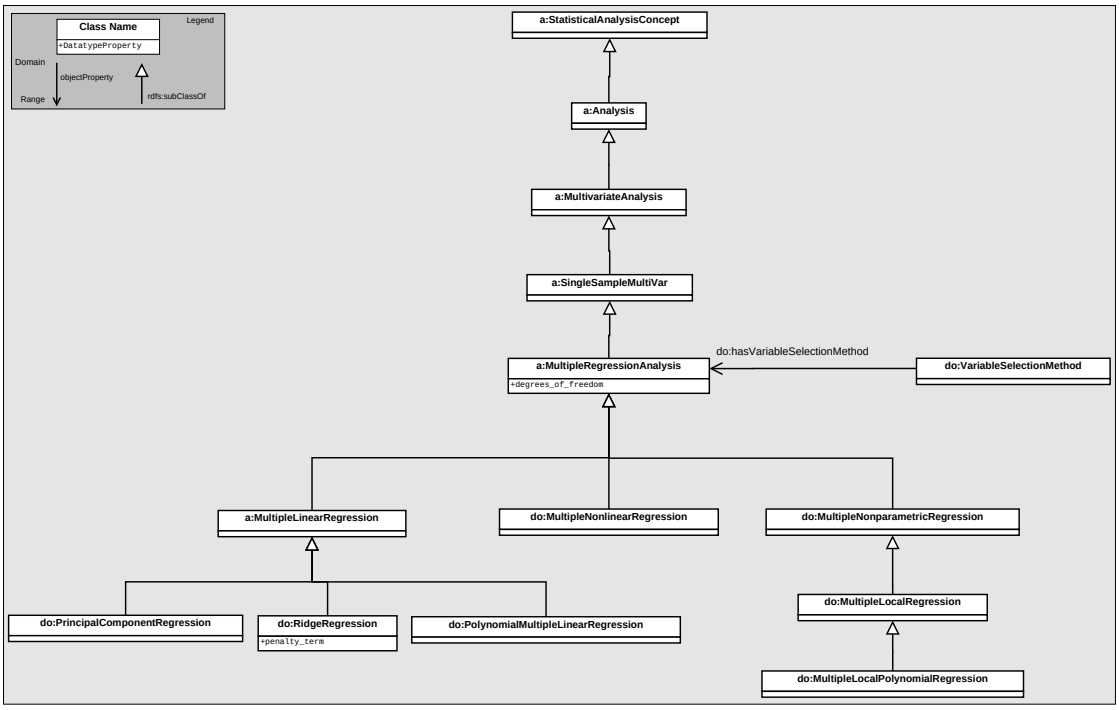


Figura E.3: Classe (a:MultipleRegressionAnalysis).

As classes (do:PrincipalComponentRegression) e (do:RidgeRegression) são estendidas a partir da regressão linear múltipla, as quais podem ser usadas para resolver o problema de multicolinearidade, quando uma variável independente dependente de outras covariáveis na regressão múltipla. Também é estendida a classe de regressão linear múltipla polinomial.

A partir de (a:MultipleRegressionAnalysis), são estendidas as classes de regressão não-linear múltipla e regressão não-paramétrica múltipla, onde esta última inclui as subclasses de regressão local múltipla e regressão polinomial local múltipla, para modelagem, por exemplo, do método de regressão não-paramétrica *Loess* múltiplo.

Na Figura E.4 é apresentado os relacionamentos a partir da Análise de Regressão Simples e Múltipla. Foi feita a inclusão na classe (a:DataTransformation) de outros possíveis tipos de transformações que podem ser aplicados nas variáveis para análise de regressão bivariada ou múltipla, tais como logarítmica, raiz quadrada, raiz cúbica, entre outras, onde a associação é feita por meio do relacionamento (do:hasDataTransformation) com a classe (a:Variable).

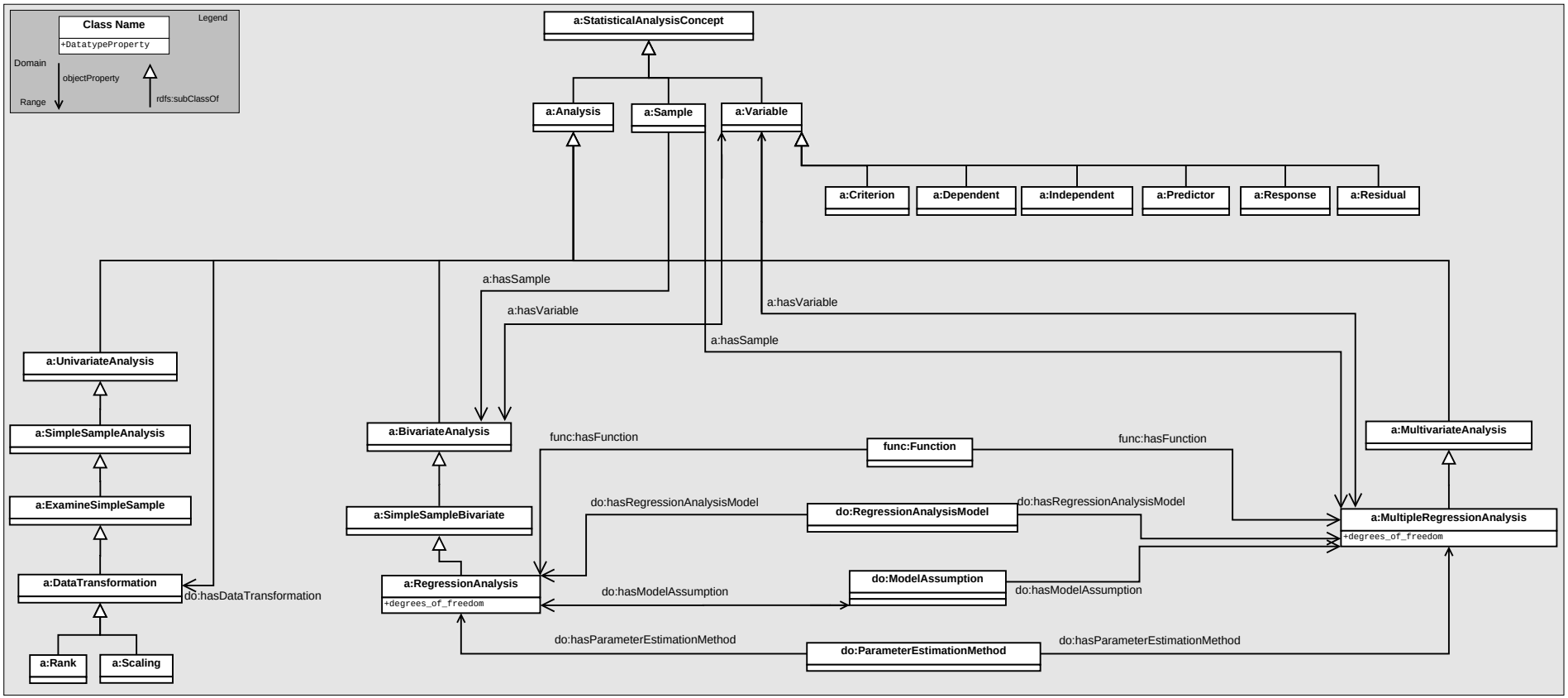


Figura E.4: Classes (a:RegressionAnalysis) e (a:MultipleRegressionAnalysis).

A extensão (Figura E.4) inclui os relacionamentos com classes para geração de conhecimento quanto à regressão aplicada, incluindo: métodos de estimação de parâmetros, modelo utilizado e suposições consideradas, funções (e grau) que podem ser ajustadas.

A Figura E.5 apresenta a extensão quanto aos métodos de estimação de parâmetros. Apesar da maioria dos métodos de regressão considerar a estimação dos parâmetros da regressão usando o método dos mínimos quadrados, em *detrending*, outros métodos podem ser usados, tais como o método da máxima vizinhança, dos momentos, entre outros, os quais apresentam variações. Também o método OLS pode apresentar variações, incluídas como instâncias, tais como mínimos quadrados ponderados, mínimos quadrados perpendiculares, entre outras.

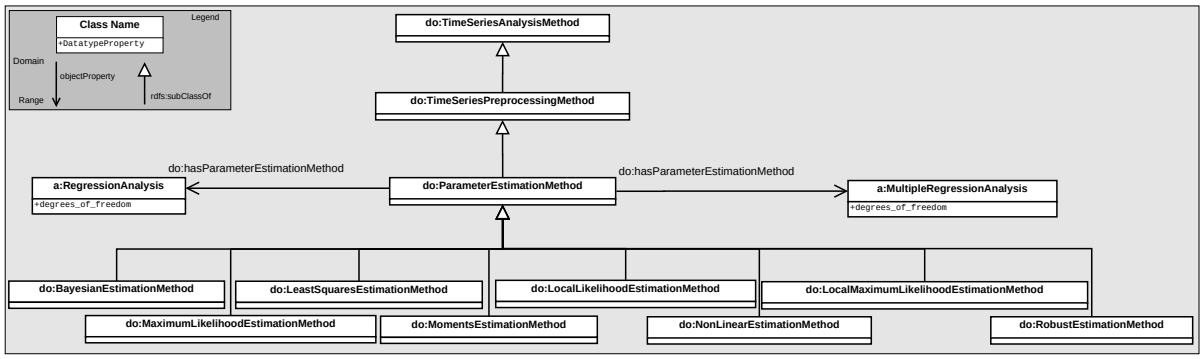


Figura E.5: Classe (do:ParameterEstimationMethod.)

A Figura E.6 apresenta qual modelo é ajustado na regressão, por exemplo, se foi usado um modelo linear ou linear por partes (*piecewise model*) ou um modelo logarítmico não-linear.

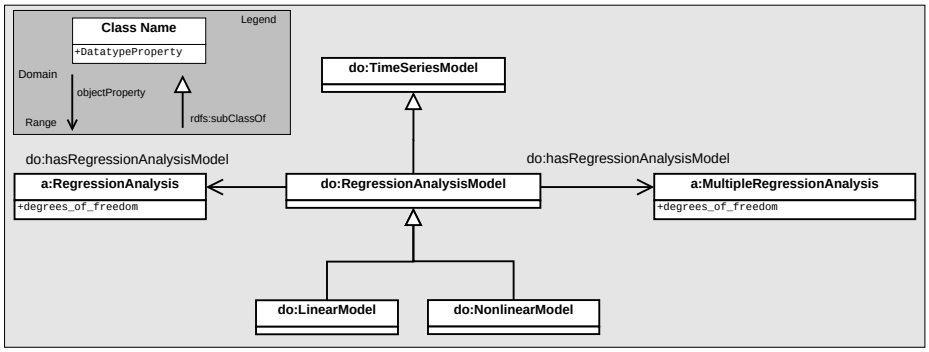


Figura E.6: Classe (do:RegressionAnalysisModel).

Para se declarar qual função é ajustada na regressão, incluindo funções trigonométricas, algébricas, *spline*, entre outras, as Figuras E.7 e E.8 mostram o reuso e extensão a partir de declarações de ontologias SWEET da classe (func:Function), sendo também reutilizada a propriedade (mrela:hasFunction) e instâncias.

A Figura E.9 mostra as medidas do sumário estatístico das séries temporais usadas na regressão, tais como média, variância e desvio-padrão, obtidas a partir do reuso de declarações SWEET.

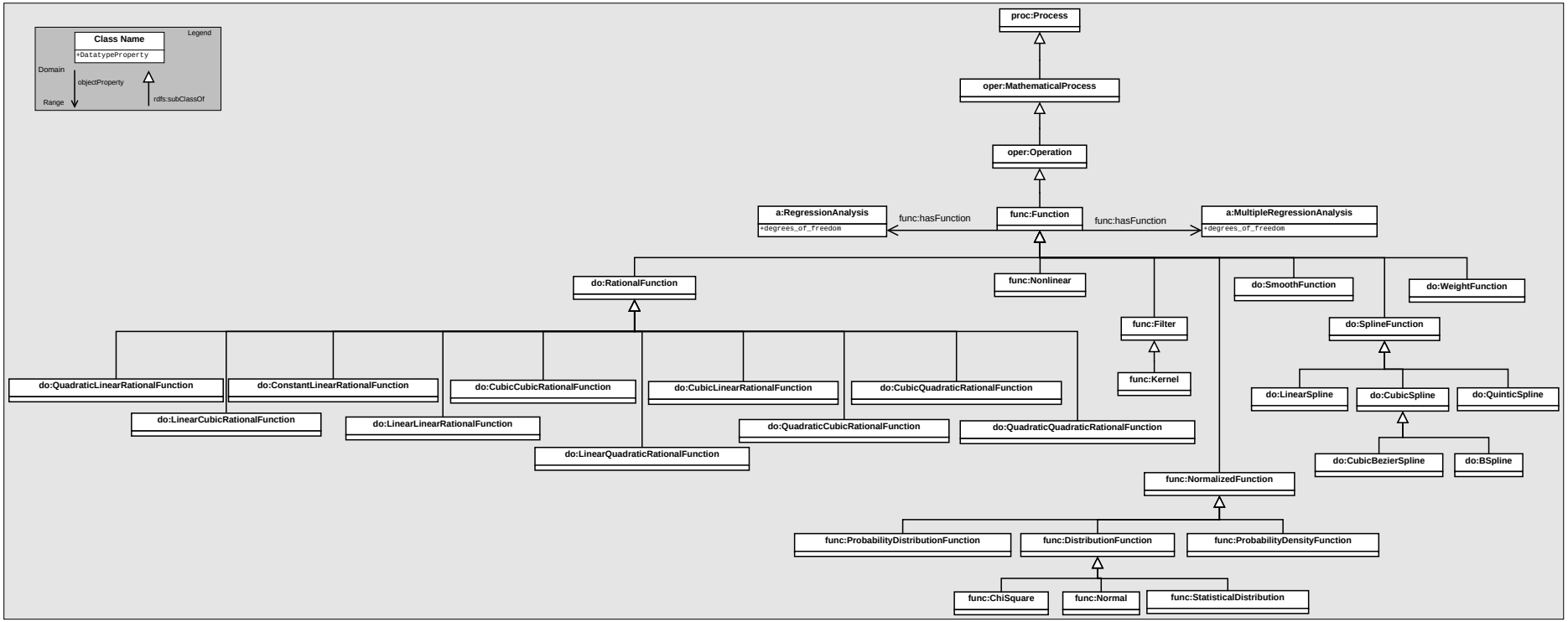


Figura E.7: Classe (func:Function).

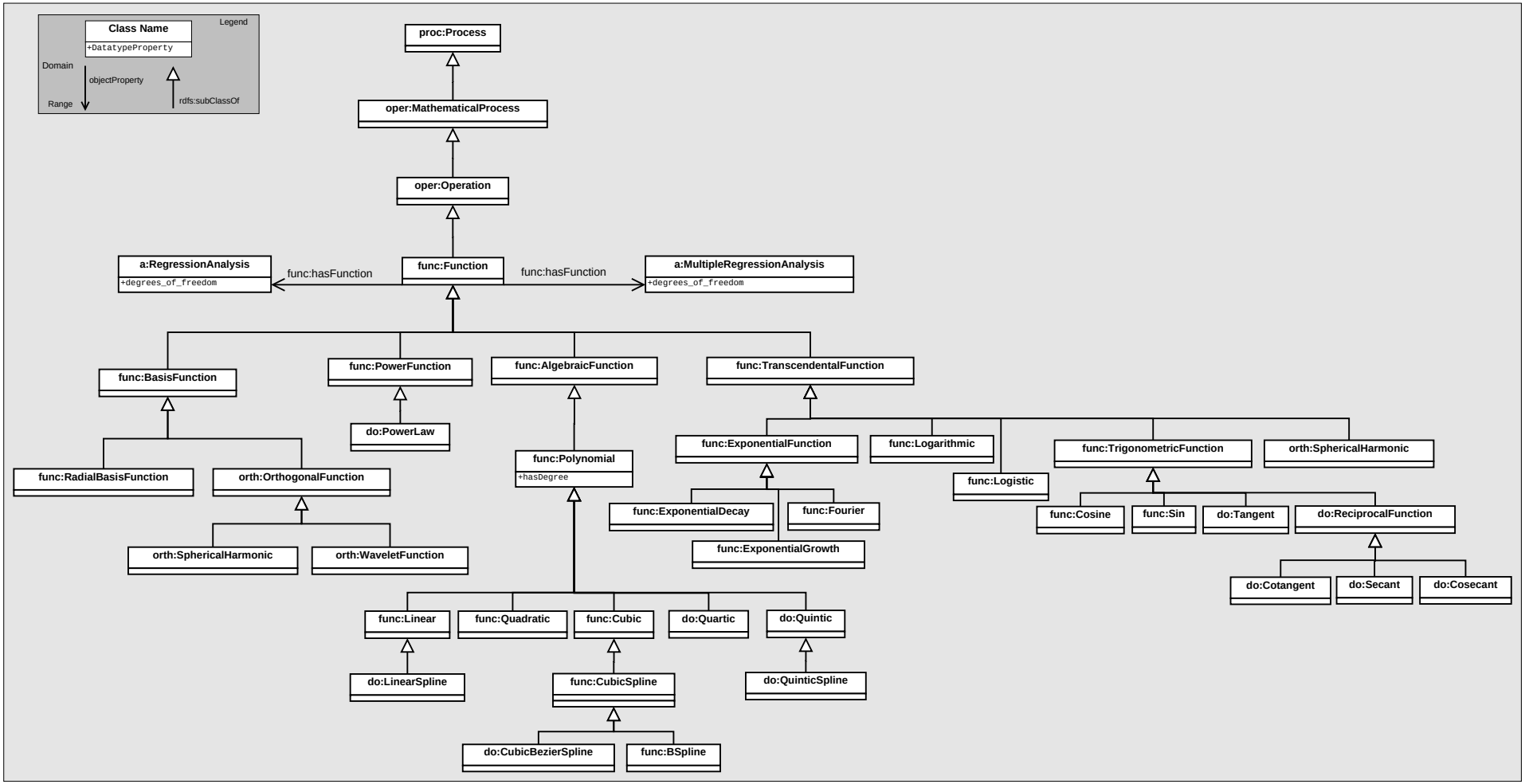


Figura E.8: Classe (func:Function) (Cont.)

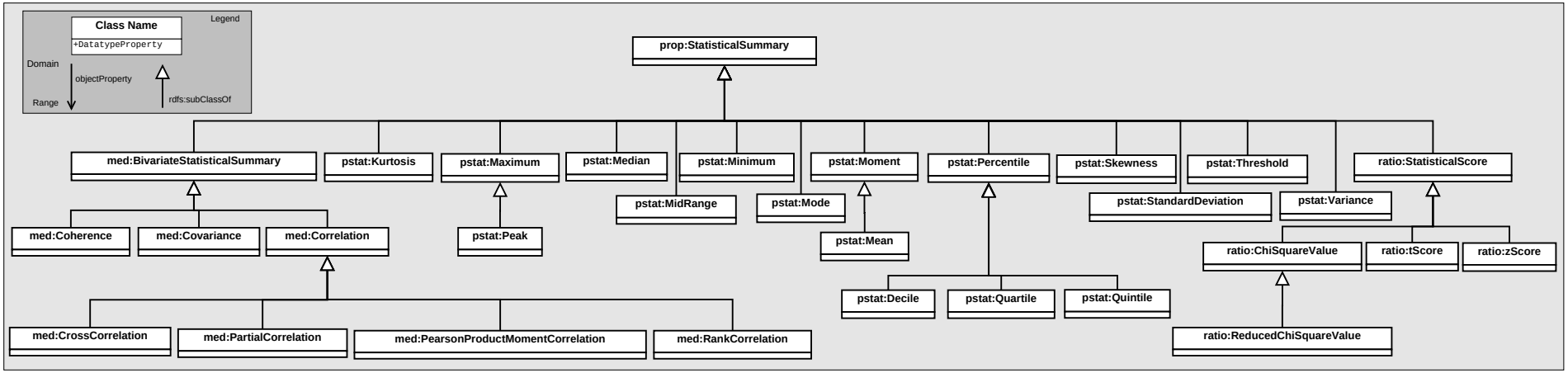


Figura E.9: Classe (prop:StatisticalSummary).

A Figura E.10 apresenta as classes de testes de hipótese e a Figura E.11 mostra a classe relacionada à qualidade de ajuste do modelo de regressão (*goodness-of-fit*), contendo classes referentes a *RSquare*, *AdjustedRSquare*, *RootMeanSquaredError* e *SumOfSquares-DueToError*.

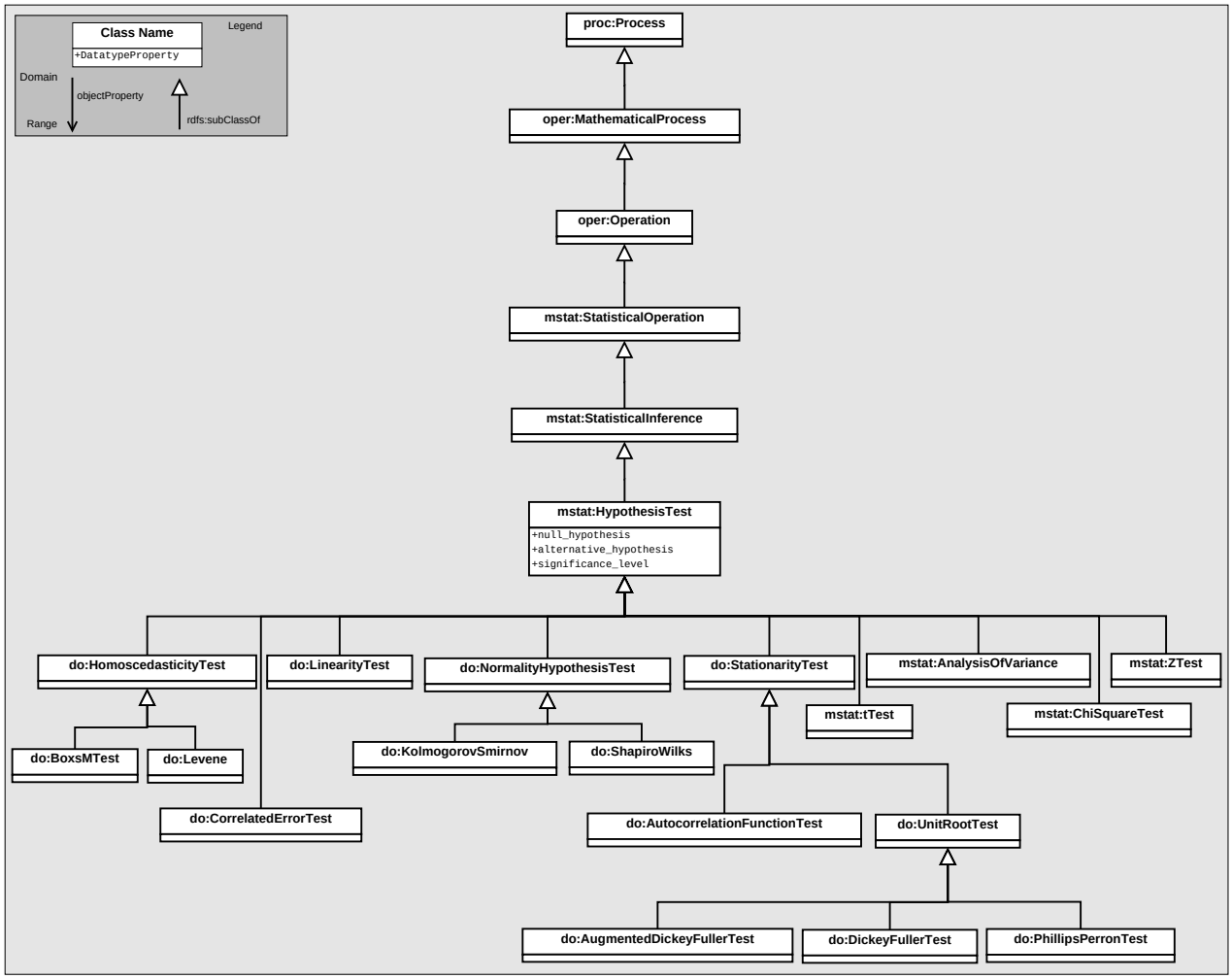


Figura E.10: Classe (mstat:HypothesisTest).

A Figura E.12 apresenta as estatísticas dos métodos e/ou sua aplicabilidade, incluindo a Estatística Paramétrica, Não-Paramétrica e Semi-Paramétrica.

A estimação da tendência de forma não-paramétrica é relacionada ao uso de um método de regressão não-paramétrica, baseado em alguma forma de suavização, conforme citam Chandler e Scott [95]. Tais métodos são apresentados na Figura E.13, os quais apresentam a propriedade de largura da banda (*bandwidth*).

Na Figura E.14, a partir da classe (do:NonparametricSimpleRegression), são criadas as classes (do:LocalRegression), (do:PenalizedLeastSquaresRegression), equivalente a (do:-SmoothingSpline) e (do:SplineRegression), esta última contém como subclasse a regressão spline cúbica.

A subclasse (do:LocalRegression) herda axiomas da classe (a:RegressionAnalysis) e possui o axioma (do:hasWeightFunction some do:WeightFunction) e contém (do:KernelRegression), (do:LocalPolynomialRegression) e (do:NearestNeighborLinearRegression) como subclasses. Tais métodos de regressão local apresentam uma localização paramétrica local

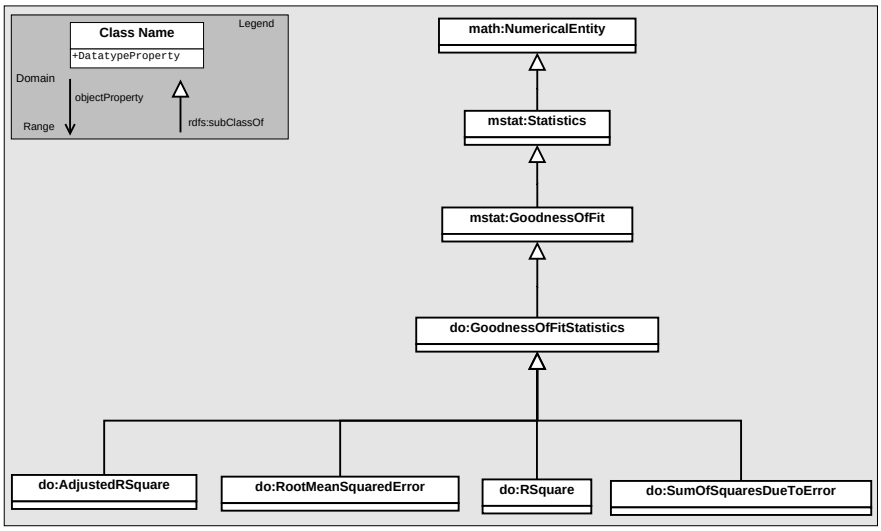


Figura E.11: Classe (mstat:GoodnessOfFit).

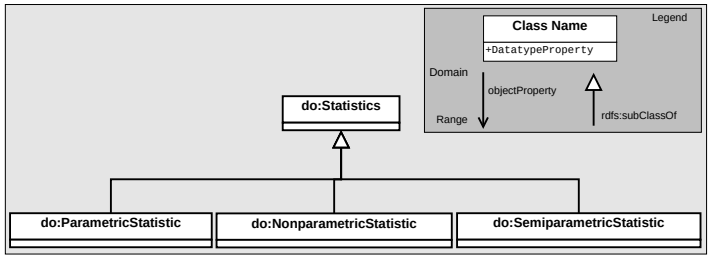


Figura E.12: Classe (do:Statistics).

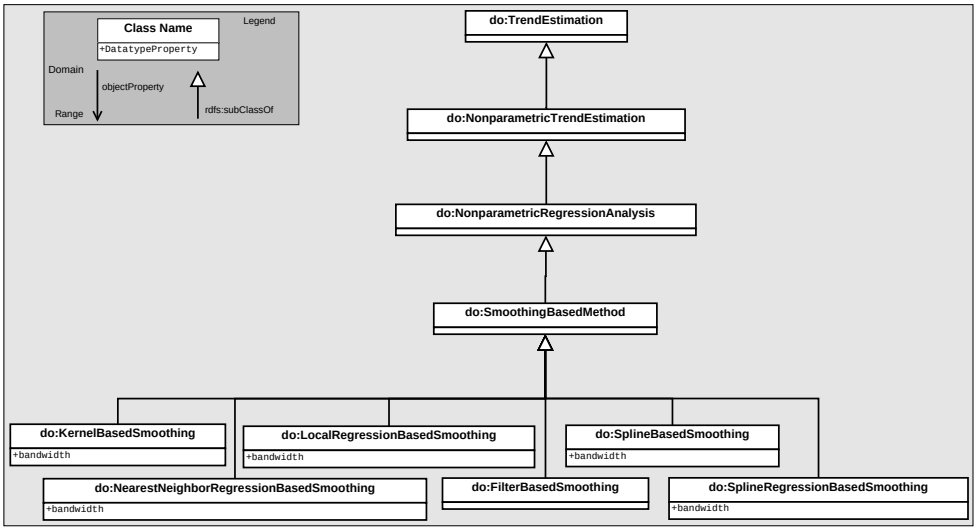


Figura E.13: Classe (do:SmoothingBasedMethod).

que os diferem de outros métodos [158].

A classe (do:KernelRegression) contém os axiomas (do:hasLocalPolynomialFunction some func:Polynomial) e (do:hasKernelFunction value Weight_Function), a qual é estendida em (do:LocalLinearKernelEstimator), (do:NadarayaWatsonKernelEstimation) e (do:PriestleyChaoKernelEstimator).

A classe (do:LocalPolynomialRegression) tem como axioma (do:hasLocalPolynomial-Function some func:Polynomial), sendo estendida em (do:LocallyWeightedPolynomialRegression), que tem como axioma (do:hasKernelFunction value do:Tricube_Kernel_Weight_-Function) e é estendida em (do:LocallyWeightedLinearRegression) como classe equivalente a (do:LowessRegression) que tem como axioma (do:hasLocalPolynomialFunction value do:LinearFunction), pois a localização paramétrica do método Lowess é ajustada localmente por um polinômio de primeiro grau e (do:LocallyWeightedQuadraticRegression) como classe equivalente a (LoessRegression) que tem como axioma (do:hasLocalPolynomialFunction value do:Quadratic_Function), onde a localização paramétrica do método Loess é ajustada localmente por um polinômio de segundo grau, conforme [203].

A Figura E.14 também apresenta a associação dos métodos de suavização com a respectiva análise de regressão, por meio da propriedade (do:hasAnalysis) e sua subclasse (do:hasLocalAnalysis). O método de suavização baseada em filtro apresenta o seguinte axioma para associação com as respectivas classes de filtros lineares e não-lineares: (do:has-Filter some do:LinearFilter and do:hasFilterDesign value do:Low_Pass) or (do:hasFilter some do:NonLinearFilter).

Quanto ao uso de filtros, a Figura E.15 apresenta a classe (do:Filtering), onde os filtros encontram-se divididos em filtros lineares e não-lineares.

O conhecimento sobre a linearidade do filtro contribui para a tomada de decisão quanto ao seu uso devido às características das séries temporais. Caso as séries apresentem algum componente de evento extremo como *jumps*, e estes são considerados como parte da tendência, o uso de filtros não-lineares, segundo Meinl [183], é considerado um método adequado para modelar tais características.

Filtros lineares são classificados conforme a resposta do impulso em *Finite Impulse Response - FIR Filter* e *Infinite Impulse Response - IIR Filter*. Os filtros lineares FIR possuem um método de implementação não-recursivo, usando convolução e os filtros IIR são implementados de forma recursiva. A Figura E.16 apresenta o projeto dos filtros lineares (*BandStopFilter*, *BandPassFilter*, *LowPassFilter* e *HighPassFilter*) e a Figura E.17 mostra as classes quanto à forma de implementação dos mesmos.

A Figura E.18 apresenta a classe de filtros FIR lineares e a Figura E.19 apresenta os filtros lineares IIR, onde os mesmos podem ter implementações de passa alta ou baixa frequência, como o filtro *Butterworth*. Da mesma forma, o mesmo método pode ter implementações no domínio do tempo ou da frequência. Um filtro linear pode ter implementação de forma não-recursiva como o filtro de médias móveis ou ser implementado de forma recursiva, como o filtro de médias móveis exponencial.

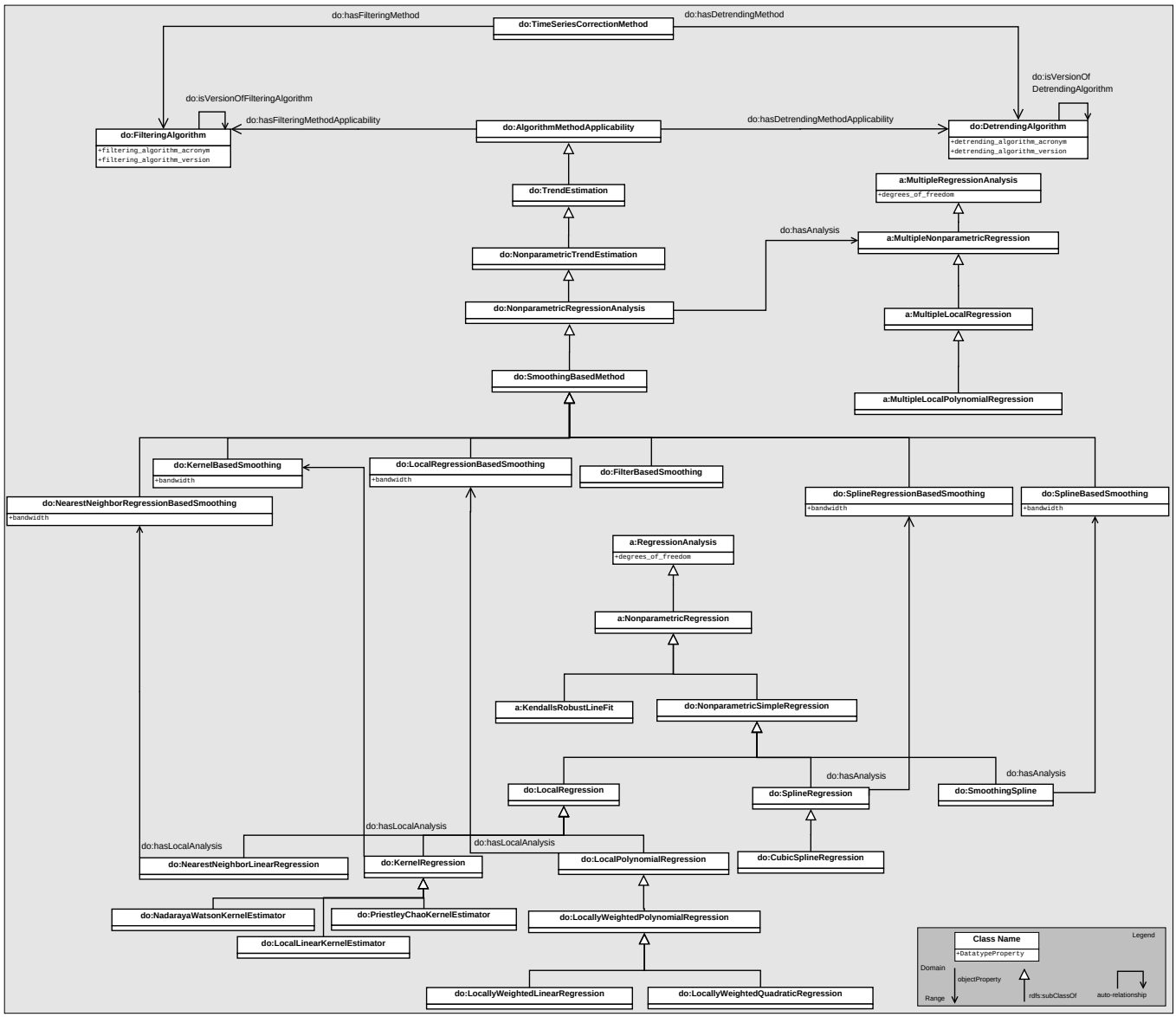


Figura E.14: Classe (do:NonParametricTrendEstimation).

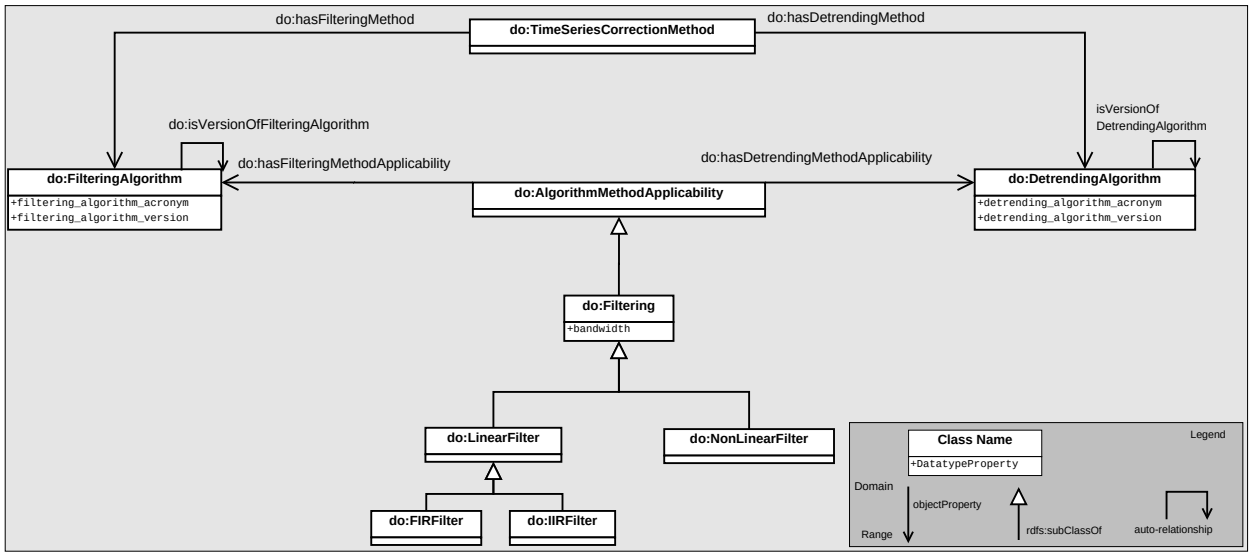


Figura E.15: Classe (do:Filtering).

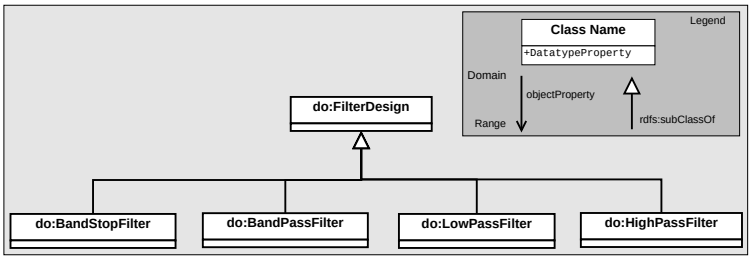


Figura E.16: Classe (do:FilterDesign).

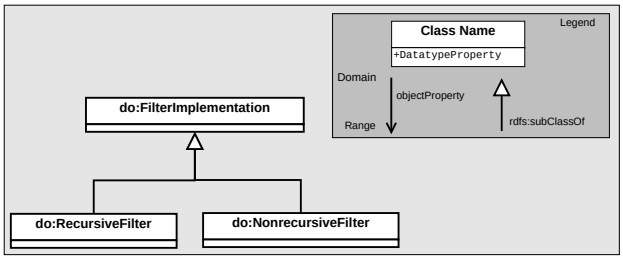


Figura E.17: Classe (do:FilterImplementation).

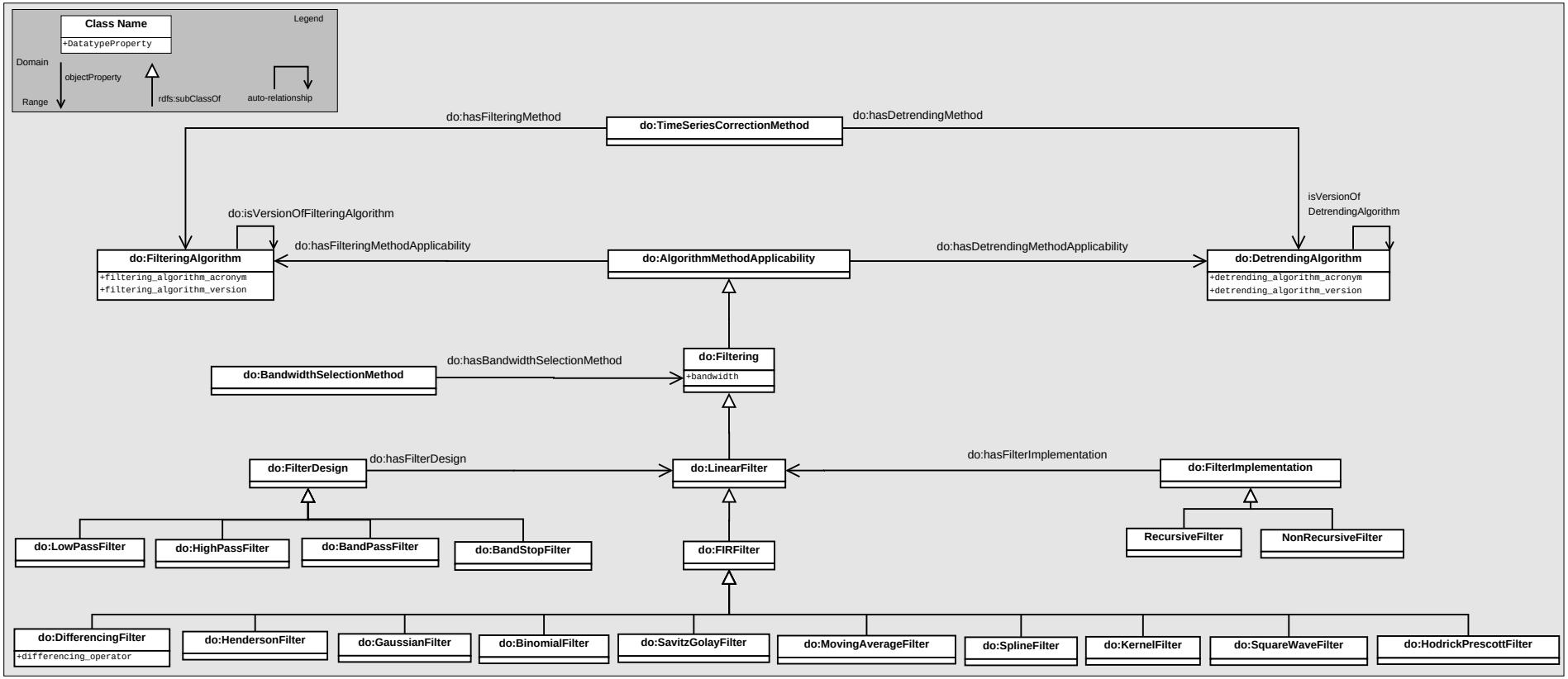


Figura E.18: Classe (do:LinearFilter).

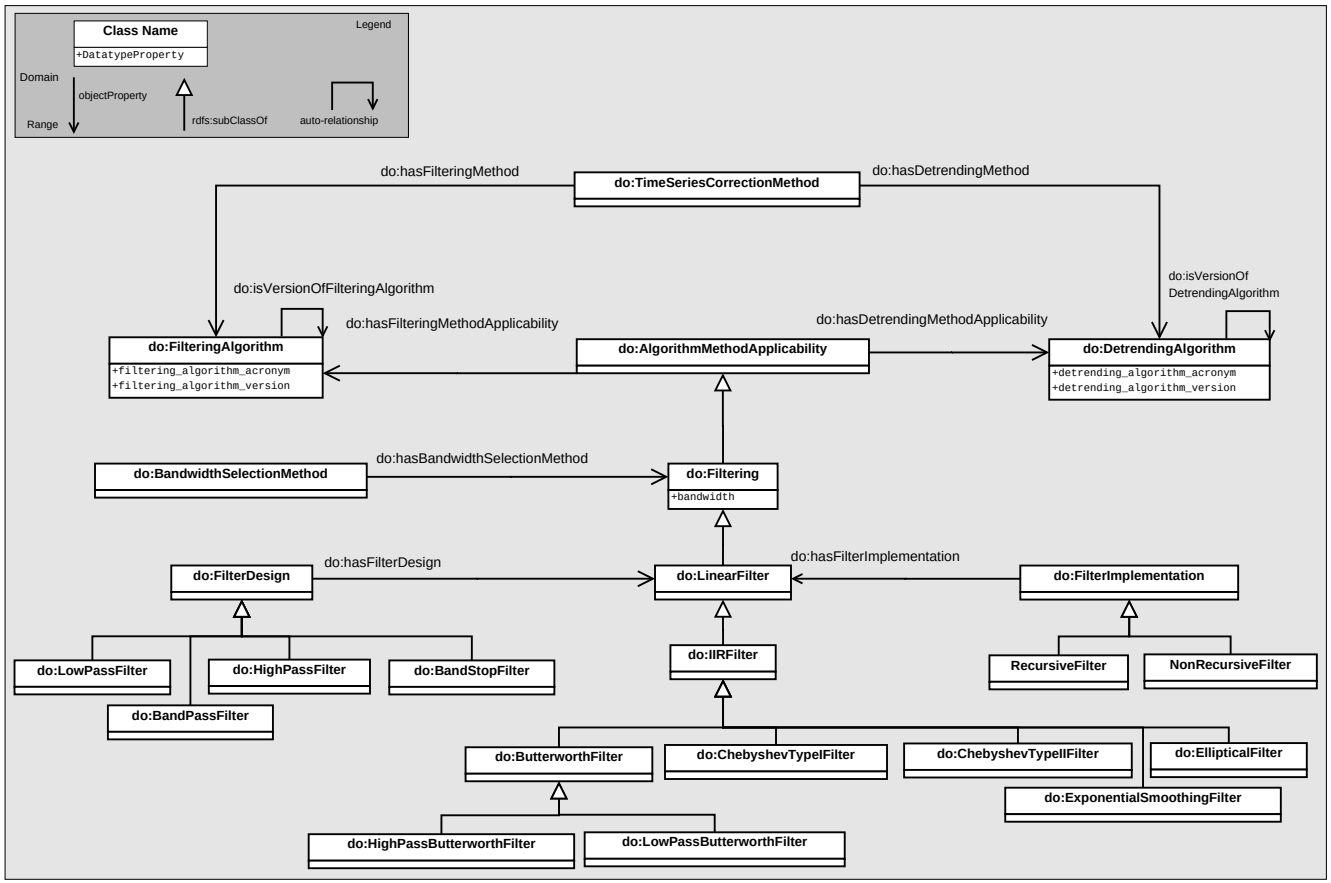


Figura E.19: Classe (do:IIRFilter).

O conhecimento sobre o projeto do filtro é relevante ao passo de *detrending*, visto que filtros passa baixa frequência são filtros de suavização, os quais bloqueiam componentes de alta frequência (ruído) e deixam passar componentes de baixa frequência, permitindo melhor identificação da tendência. Outro caso são os filtros que permitem a passagem de alta frequência, os quais bloqueiam componentes de baixa frequência (tendências) e permitem passar componentes de alta frequência (ruído).

Os filtros não-lineares (Figura E.20) não são implementados usando uma convolução linear, mas utilizando alguma forma de ordenação dos pares vizinhos (*ranking*). Um dos mais utilizados para *detrending* é o filtro de medianas móveis (do:MedianFilter), que faz um *ranking*, considerando cinquenta *percentile* dos pontos vizinhos [183, 212].

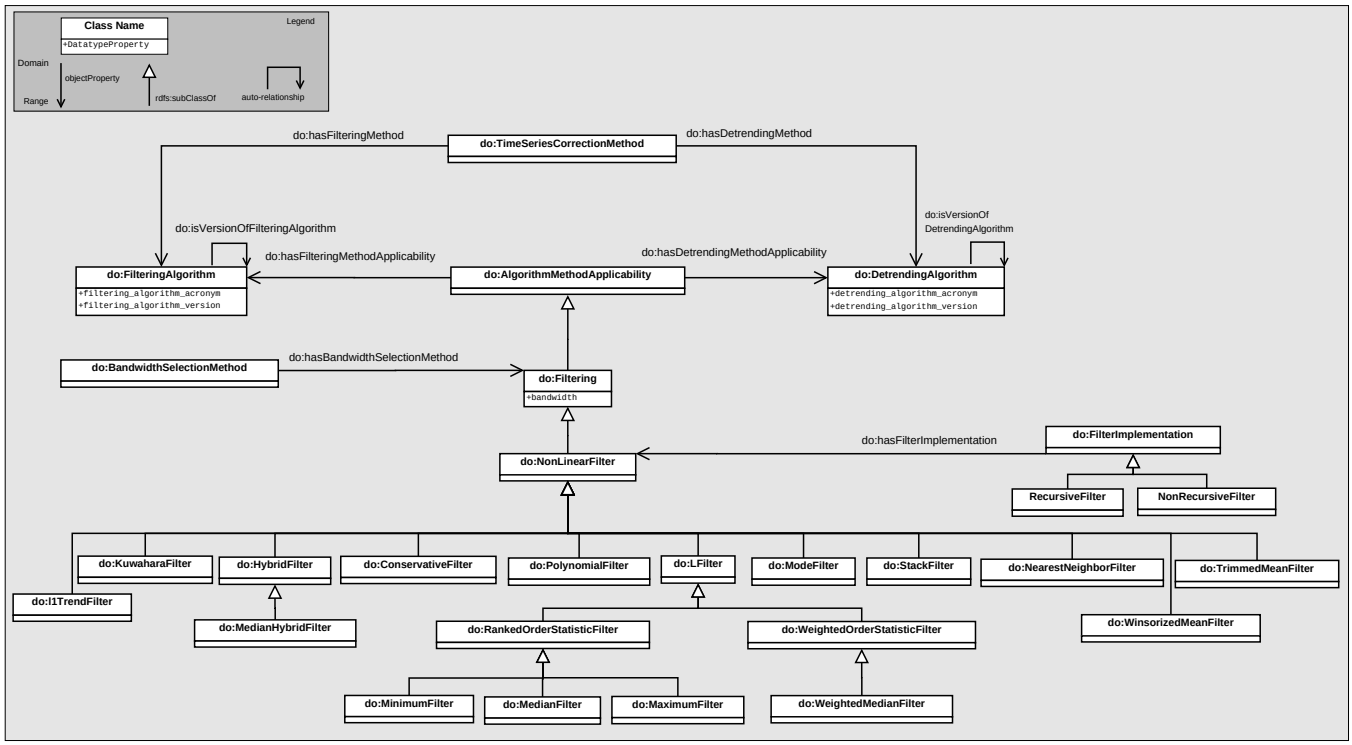


Figura E.20: Classe (do:NonLinearFilter).

Métodos de filtros necessitam de uma forma de seleção da largura da banda (Figura E.18), onde o pesquisador pode escolher o valor desse parâmetro de forma subjetiva, dado seu conhecimento sobre o sistema, ou usar um método para escolha da largura da banda, como o método de Cross-Validação [95], entre outros.

Para modelagem do respectivo paradigma/linguagem de programação utilizada pelos softwares de *detrending* (Figura E.21), a definição das classes é baseada na nomenclatura ACM, o que justifica a exceção para estas classes estarem definidas no plural.

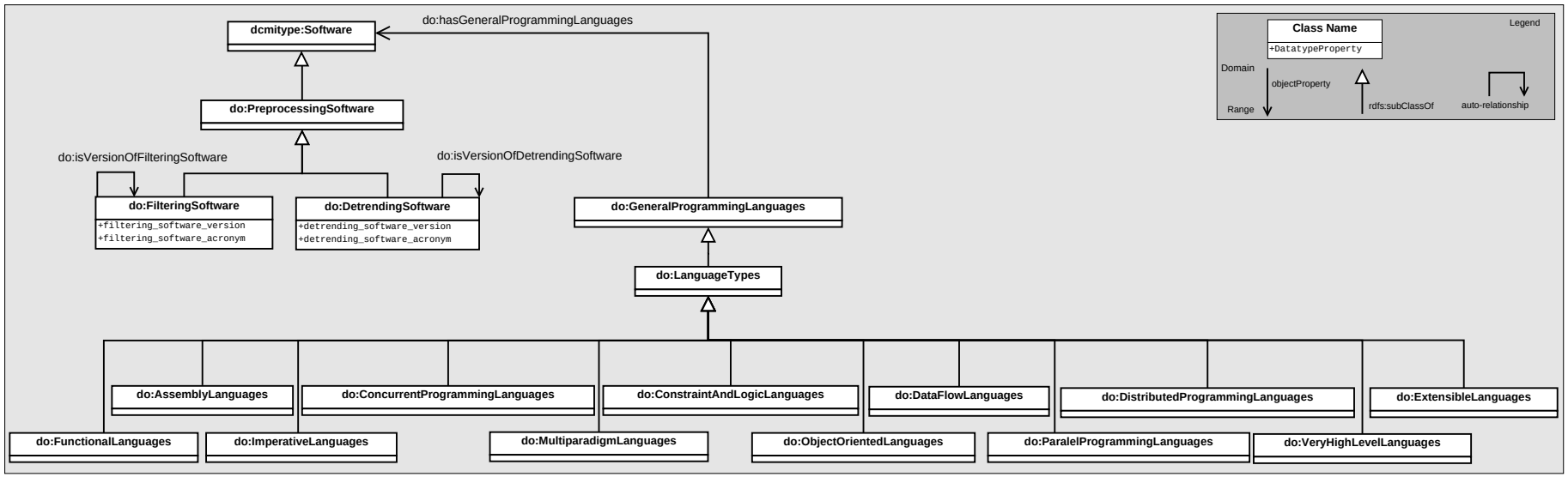


Figura E.21: Classe (do:GeneralProgrammingLanguages).

A classificação dos algoritmos e softwares é feita com base no tipo de método e sua aplicabilidade. As Figuras E.22 e E.23 apresentam, respectivamente, a classificação dos algoritmos e softwares de filtros que podem ser usados para correção do ruído de alta frequência das séries temporais. As propriedades de dados consideradas são o acrônimo e a versão dos mesmos.

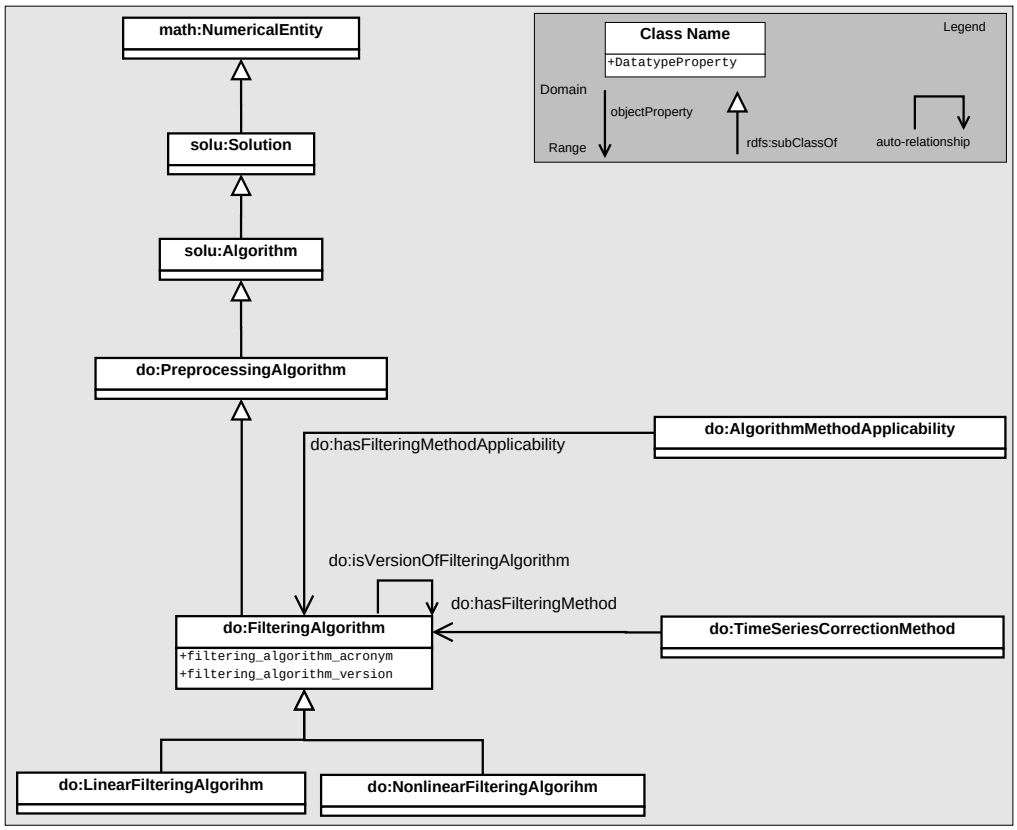


Figura E.22: Classe (do:FilteringAlgorithm).

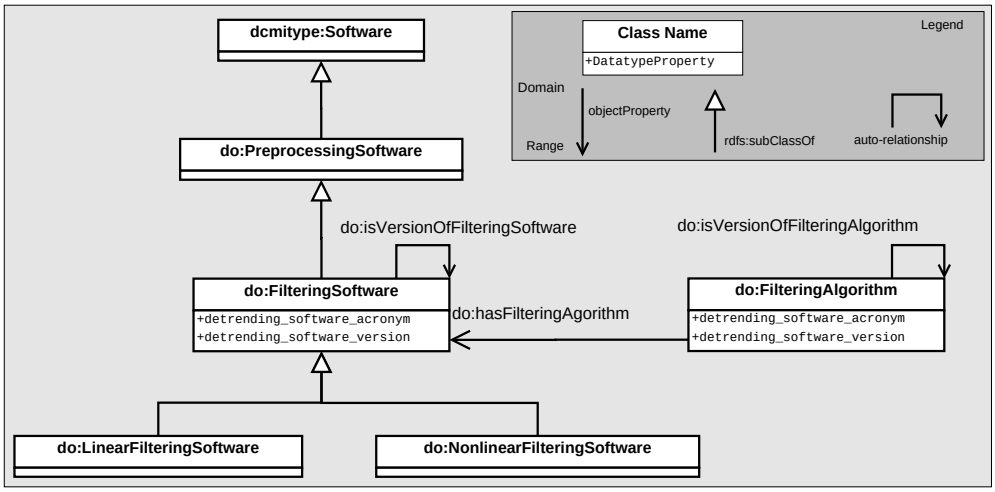


Figura E.23: Classe (do:FilteringSoftware).

As Figuras E.24, E.25, E.26 e E.27 apresentam a classificação dos algoritmos e softwares de forma paramétrica e não-paramétrica, respectivamente.

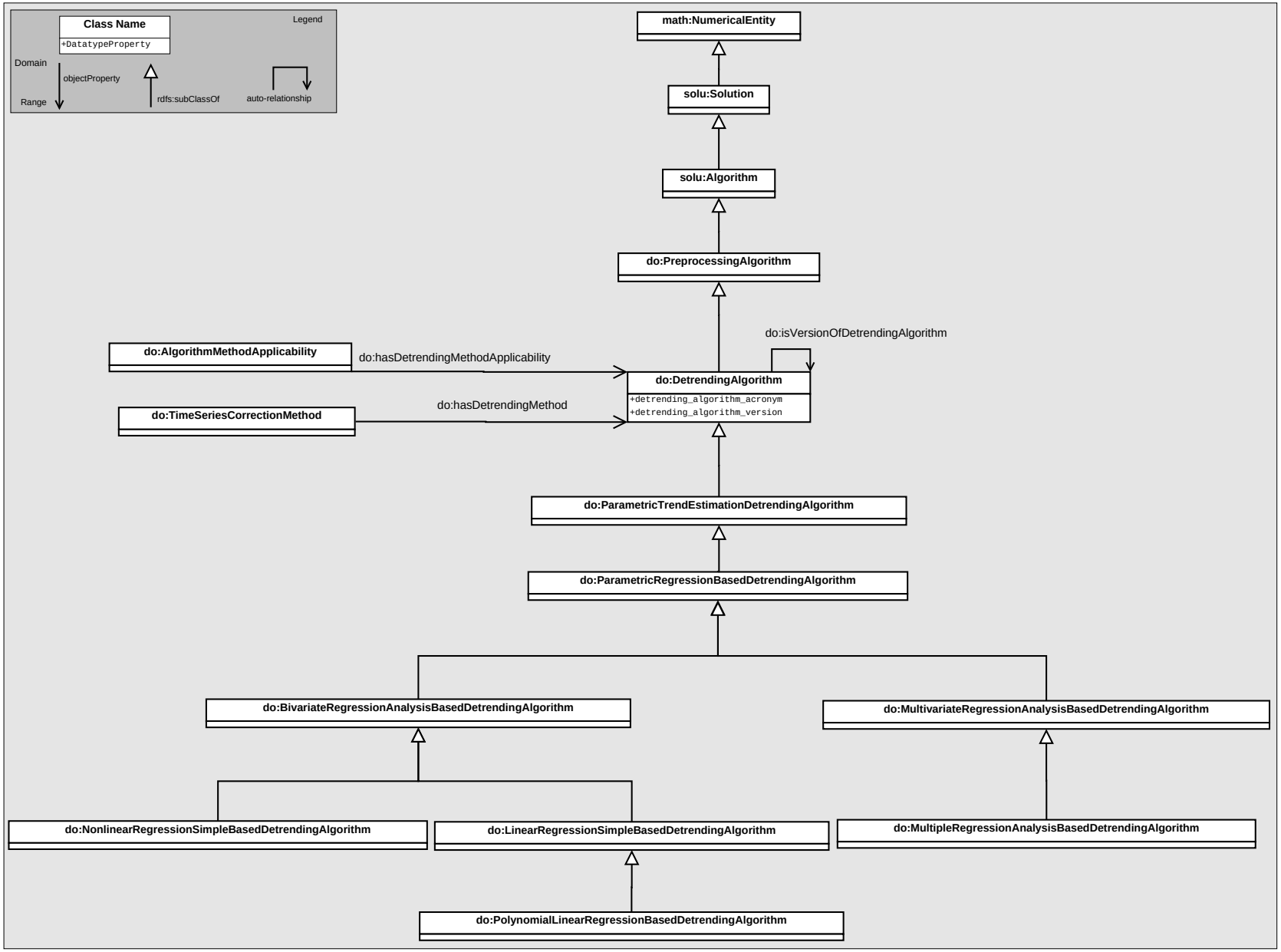


Figura E.24: Classe (do:ParametricTrendEstimationDetrendingAlgorithm).

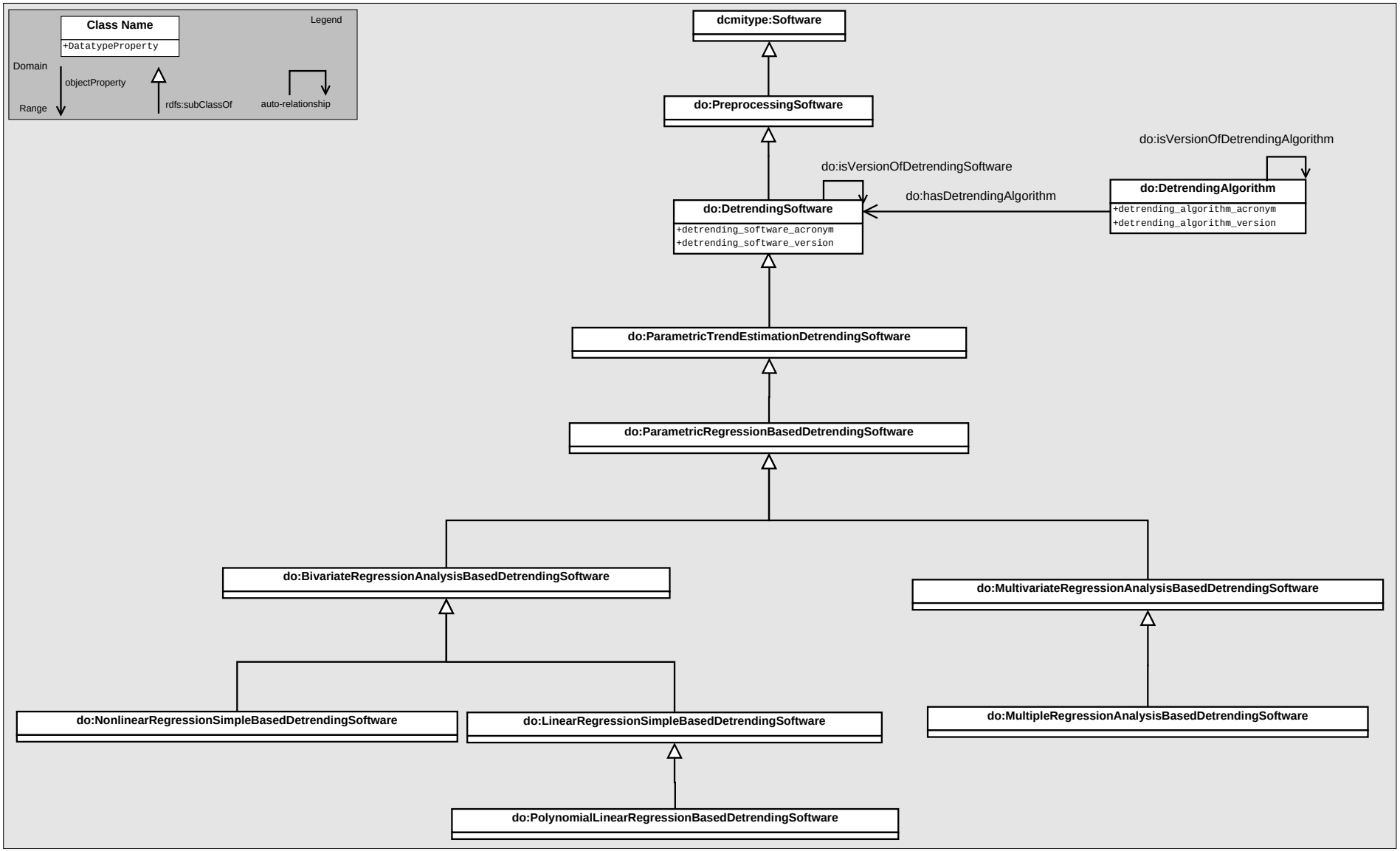


Figura E.25: Classe (do:ParametricTrendEstimationDetrendingSoftware).

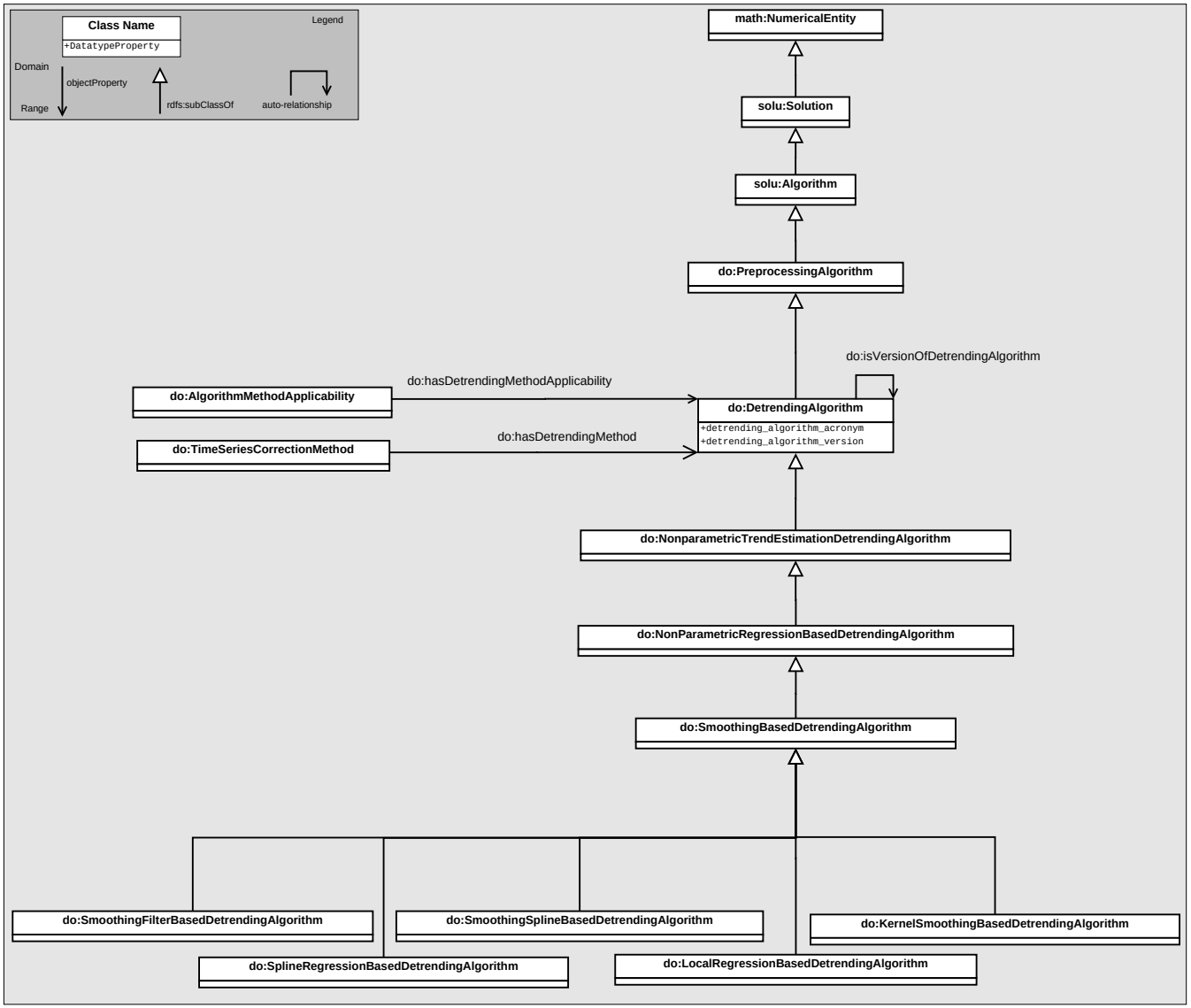


Figura E.26: Classe (do:NonParametricTrendEstimationDetrendingAlgorithm).

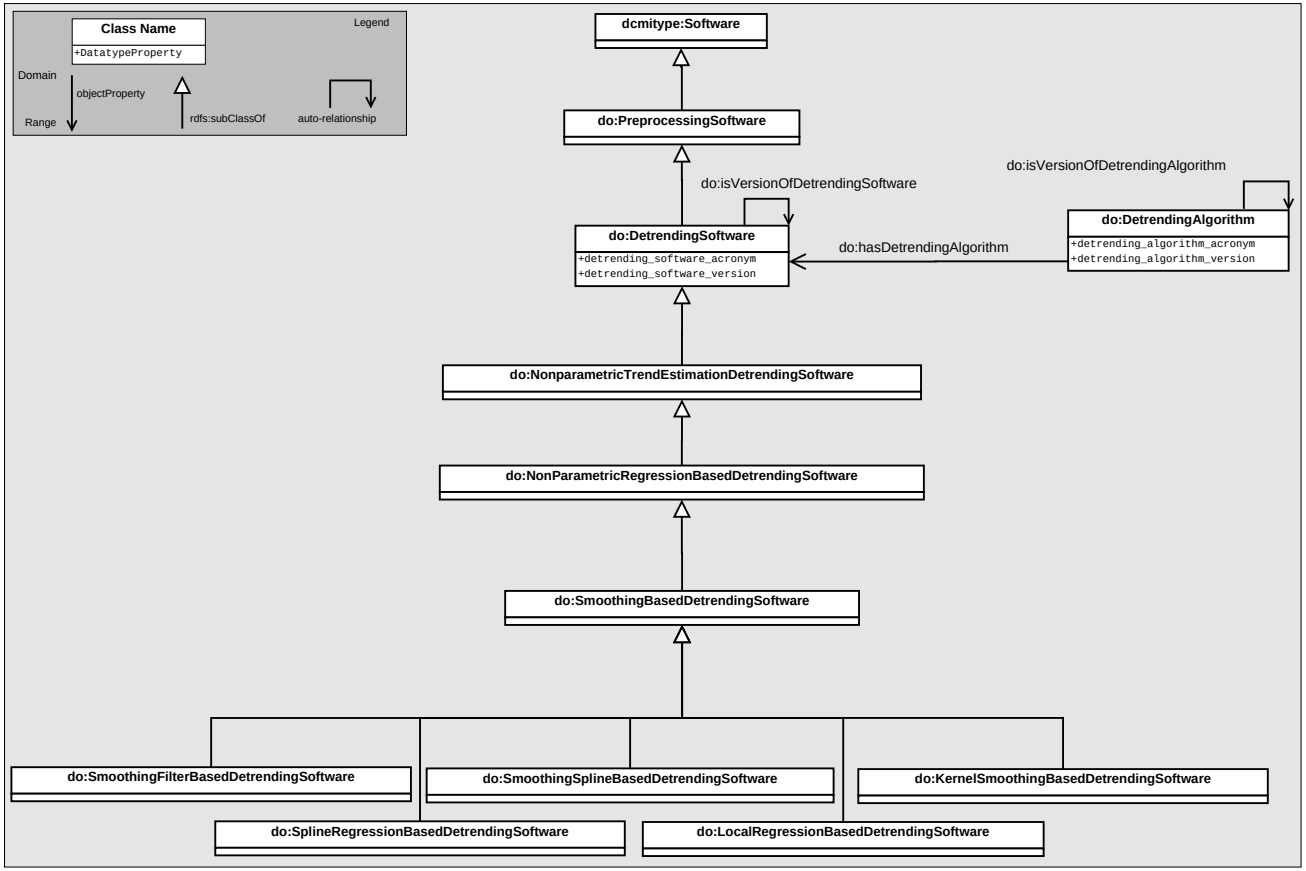


Figura E.27: Classe (do:NonparametricTrendEstimationDetrendingSoftware).

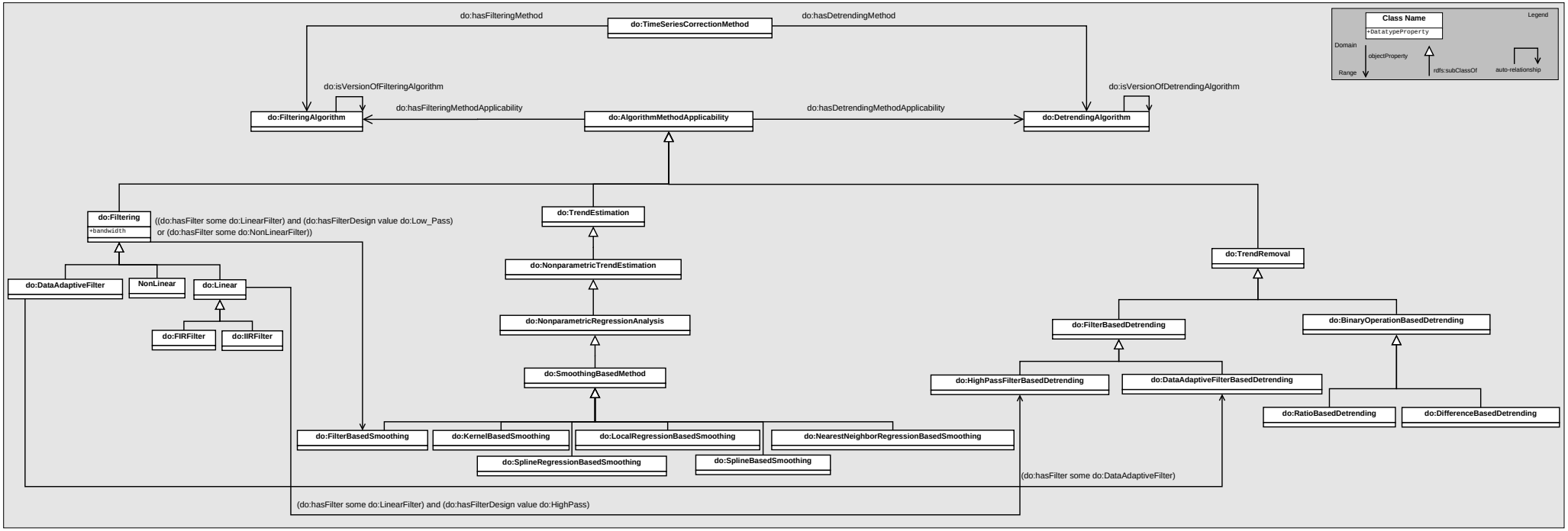


Figura E.28: Classes (do:TrendRemoval) e (do:Filtering).

A Figura E.28 mostra os relacionamentos, a partir dos métodos de estimação e remoção de tendências usando métodos de filtros. Por exemplo, *detrending* baseado em filtro passa alto é associado com a classe (do:Filtering) por meio do axioma: (do:hasFilter some do:LinearFilter and do:hasFilterDesign value do:High_Pass).

Quanto à remoção de tendências, a Figura E.29 descreve como esse procedimento pode ser realizado. Na ontologia em questão, as tarefas de estimação e remoção de tendências são modeladas de forma separada, sendo apresentado como a tendência é removida. A partir da classe (do:TrendRemoval), são criadas as subclasses (do:FilterBasedDetrending) e subclasse (do:HighPassFilteringBasedDetrending) e (do:BinaryOperationBasedDetrending) e subclasses (do:DifferenceBasedDetrending) e (do:RatioBasedDetrending).

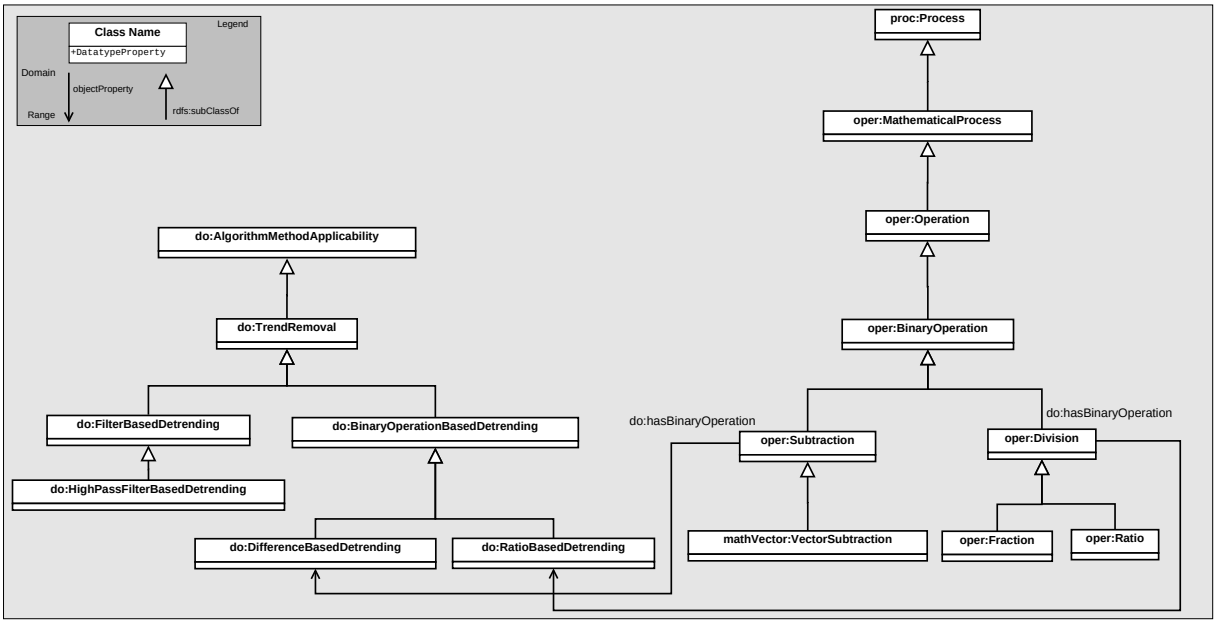


Figura E.29: Classe (do:TrendRemoval).

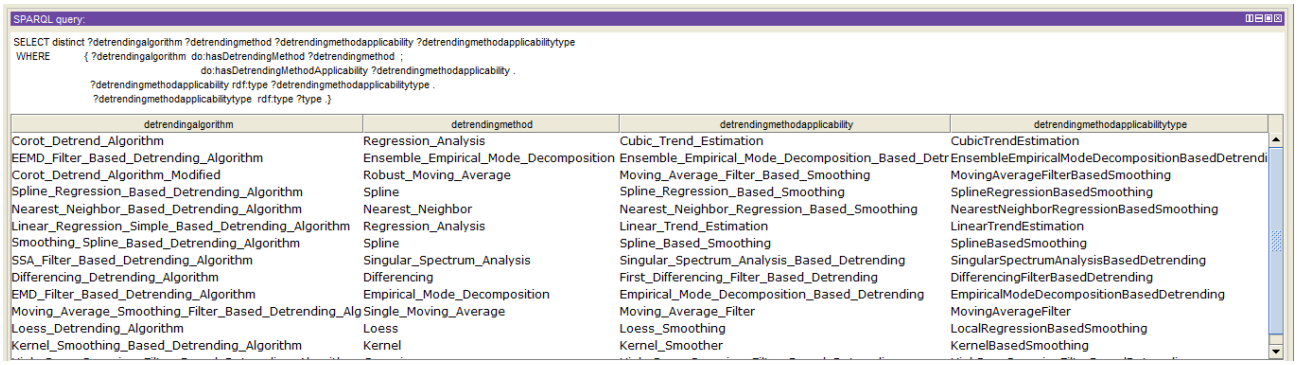
Quanto ao *detrending* baseado em filtro passa alta frequência, ao considerar tendências como estocásticas, por exemplo, estas podem ser removidas pelo método de *Differencing* [223], que é um filtro passa alta frequência, bloqueando a baixa frequência das séries temporais (tendências).

O método de *detrending* baseado em operação binária é utilizado, no caso de tendências determinísticas ou que foram suavizadas, onde, após a estimação da tendência, a mesma é removida conforme o modelo de decomposição das séries temporais considerado. Se o modelo de decomposição considerado for aditivo, a tendência estimada é subtraída a partir das séries temporais (do:DifferenceBasedDetrending), gerando dados corrigidos de tendência (*detrended*). Se o modelo de decomposição das séries for considerado multiplicativo, a tendência estimada é dividida, a partir das séries temporais (do:RatioBasedDetrending), também gerando dados corrigidos de tendência. Em qualquer caso, é feita a associação com a respectiva operação binária, a partir do reuso de SWEET, conforme a Figura E.29.

Na sequência são apresentadas consultas na ontologia DO.

Consultas na Ontologia DO

As Figuras E.30 a E.52 apresentam consultas na ontologia DO desenvolvidas na linguagem SPARQL.

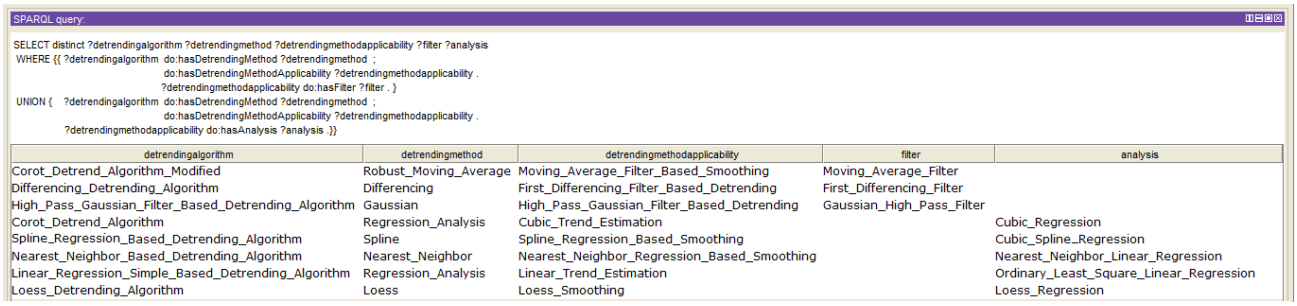


SPARQL query:

```
SELECT distinct ?detrendingalgorithm ?detrendingmethod ?detrendingmethodapplicability ?detrendingmethodapplicabilitytype
 WHERE        { ?detrendingalgorithm  do:hasDetrendingMethod ?detrendingmethod  ;
                                      do:hasDetrendingMethodApplicability ?detrendingmethodapplicability .
                ?detrendingmethodapplicability rdf:type ?detrendingmethodapplicabilitytype .
                 ?detrendingmethodapplicabilitytype  rdf:type ?type .}
```

detrendingalgorithm	detrendingmethod	detrendingmethodapplicability	detrendingmethodapplicabilitytype
Corot_Detrend_Algorithm	Regression_Analysis	Cubic_Trend_Estimation	CubicTrendEstimation
EEMD_Filter_Based_Detrending_Algorithm	Ensemble_Empirical_Mode_Decomposition	Ensemble_Empirical_Mode_Decomposition_Based_Detr	EnsembleEmpiricalModeDecompositionBasedDetrendi
Corot_Detrend_Algorithm_Modified	Robust_Moving_Average	Moving_Average_Filter_Based_Smoothing	MovingAverageFilterBasedSmoothing
Spline_Regression_Based_Detrending_Algorithm	Spline	Spline_Regression_Based_Smoothing	SplineRegressionBasedSmoothing
Nearest_Neighbor_Based_Detrending_Algorithm	Nearest_Neighbor	Nearest_Neighbor_Regression_Based_Smoothing	NearestNeighborRegressionBasedSmoothing
Linear_Regression_Simple_Based_Detrending_Algorithm	Regression_Analysis	Linear_Trend_Estimation	LinearTrendEstimation
Smoothing_Spline_Based_Detrending_Algorithm	Spline	Spline_Based_Smoothing	SplineBasedSmoothing
SSA_Filter_Based_Detrending_Algorithm	Singular_Spectrum_Analysis	Singular_Spectrum_Analysis_Based_Detrending	SingularSpectrumAnalysisBasedDetrending
Differencing_Detrending_Algorithm	Differencing	First_Differencing_Filter_Based_Detrending	DifferencingFilterBasedDetrending
EMD_Filter_Based_Detrending_Algorithm	Empirical_Mode_Decomposition	Empirical_Mode_Decomposition_Based_Detrending	EmpiricalModeDecompositionBasedDetrending
Moving_Average_Smoothing_Filter_Based_Detrending_Alg	Single_Moving_Average	Moving_Average_Filter	MovingAverageFilter
Loess_Detrending_Algorithm	Loess	Loess_Smoothing	LocalRegressionBasedSmoothing
Kernel_Smoothing_Based_Detrending_Algorithm	Kernel	Kernel_Smoother	KernelBasedSmoothing

Figura E.30: Consulta sobre quais métodos, a aplicabilidade e seu tipo nos algoritmos de *detrending*.

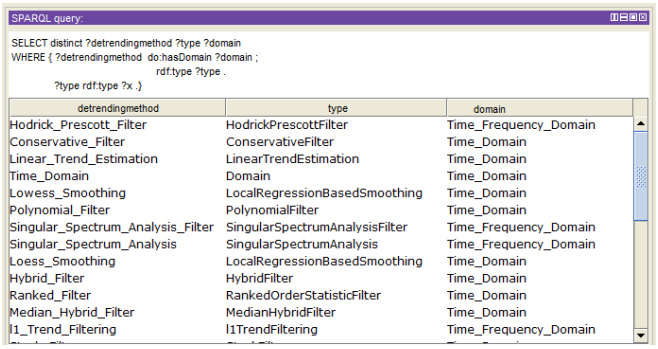


SPARQL query:

```
SELECT distinct ?detrendingalgorithm ?detrendingmethod ?detrendingmethodapplicability ?filter ?analysis
 WHERE {{ ?detrendingalgorithm  do:hasDetrendingMethod ?detrendingmethod  ;
                                do:hasDetrendingMethodApplicability ?detrendingmethodapplicability .
                               ?detrendingmethodapplicability do:hasFilter ?filter . }
 UNION {    ?detrendingalgorithm  do:hasDetrendingMethod ?detrendingmethod  ;
                                  do:hasDetrendingMethodApplicability ?detrendingmethodapplicability .
            ?detrendingmethodapplicability do:hasAnalysis ?analysis .}}
```

detrendingalgorithm	detrendingmethod	detrendingmethodapplicability	filter	analysis
Corot_Detrend_Algorithm_Modified	Robust_Moving_Average	Moving_Average_Filter_Based_Smoothing	Moving_Average_Filter	
Differencing_Detrending_Algorithm	Differencing	First_Differencing_Filter_Based_Detrending	First_Differencing_Filter	
High_Pass_Gaussian_Filter_Based_Detrending_Algorithm	Gaussian	High_Pass_Gaussian_Filter_Based_Detrending	Gaussian_High_Pass_Filter	
Corot_Detrend_Algorithm	Regression_Analysis	Cubic_Trend_Estimation		Cubic_Regression
Spline_Regression_Based_Detrending_Algorithm	Spline	Spline_Regression_Based_Smoothing		Cubic_Spline_Regression
Nearest_Neighbor_Based_Detrending_Algorithm	Nearest_Neighbor	Nearest_Neighbor_Regression_Based_Smoothing		Nearest_Neighbor_Linear_Regression
Linear_Regression_Simple_Based_Detrending_Algorithm	Regression_Analysis	Linear_Trend_Estimation		Ordinary_Least_Square_Linear_Regression
Loess_Detrending_Algorithm	Loess	Loess_Smoothing		Loess_Regression

Figura E.31: Consulta sobre os métodos, aplicabilidade e relacionamentos nos algoritmos.

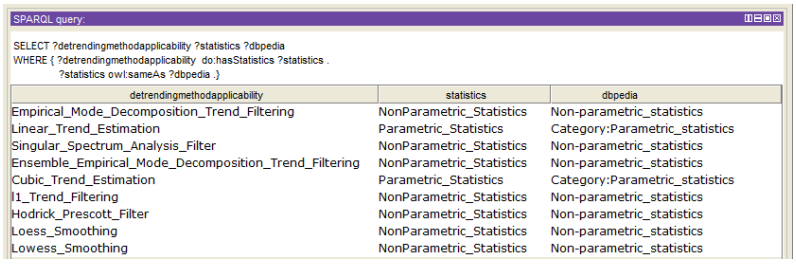


SPARQL query:

```
SELECT distinct ?detrendingmethod ?type ?domain
WHERE { ?detrendingmethod  do:hasDomain ?domain ;
                           rdf:type ?type .
        ?type rdf:type ?x .}
```

detrendingmethod	type	domain
Hodrick_Prescott_Filter	HodrickPrescottFilter	Time_Frequency_Domain
Conservative_Filter	ConservativeFilter	Time_Domain
Linear_Trend_Estimation	LinearTrendEstimation	Time_Domain
Time_Domain	Domain	Time_Domain
Lowess_Smoothing	LocalRegressionBasedSmoothing	Time_Domain
Polynomial_Filter	PolynomialFilter	Time_Domain
Singular_Spectrum_Analysis_Filter	SingularSpectrumAnalysisFilter	Time_Frequency_Domain
Singular_Spectrum_Analysis	SingularSpectrumAnalysis	Time_Frequency_Domain
Loess_Smoothing	LocalRegressionBasedSmoothing	Time_Domain
Hybrid_Filter	HybridFilter	Time_Domain
Ranked_Filter	RankedOrderStatisticFilter	Time_Domain
Median_Hybrid_Filter	MedianHybridFilter	Time_Domain
l1_Trend_Filtering	l1TrendFiltering	Time_Frequency_Domain

Figura E.32: Consulta sobre o domínio dos métodos e seu tipo.



SPARQL query:

```
SELECT ?detrendingmethodapplicability ?statistics ?dbpedia
WHERE { ?detrendingmethodapplicability  do:hasStatistics ?statistics .
        ?statistics owl:sameAs ?dbpedia .}
```

detrendingmethodapplicability	statistics	dbpedia
Empirical_Mode_Decomposition_Trend_Filtering	NonParametric_Statistics	Non-parametric_statistics
Linear_Trend_Estimation	Parametric_Statistics	Category:Parametric_statistics
Singular_Spectrum_Analysis_Filter	NonParametric_Statistics	Non-parametric_statistics
Ensemble_Empirical_Mode_Decomposition_Trend_Filtering	NonParametric_Statistics	Non-parametric_statistics
Cubic_Trend_Estimation	Parametric_Statistics	Category:Parametric_statistics
l1_Trend_Filtering	NonParametric_Statistics	Non-parametric_statistics
Hodrick_Prescott_Filter	NonParametric_Statistics	Non-parametric_statistics
Loess_Smoothing	NonParametric_Statistics	Non-parametric_statistics
Lowess_Smoothing	NonParametric_Statistics	Non-parametric_statistics

Figura E.33: Consulta sobre a Estatística dos métodos, associação com a DBpedia e sua aplicabilidade.

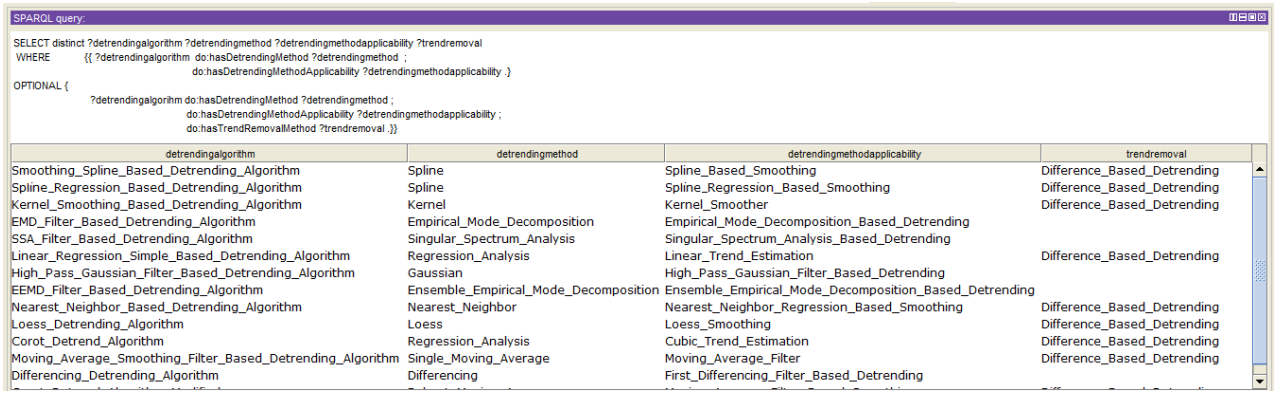


Figura E.34: Consulta sobre como é removido o componente tendência pelos algoritmos e métodos de *detrending*.

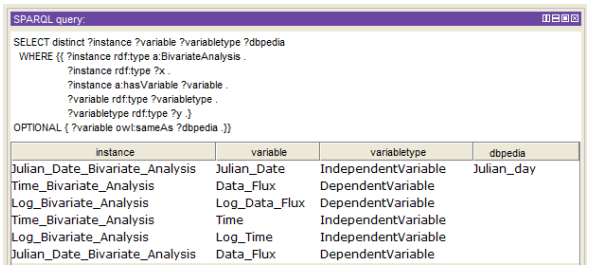


Figura E.35: Consulta sobre quais variáveis e seu tipo estão envolvidas na regressão.

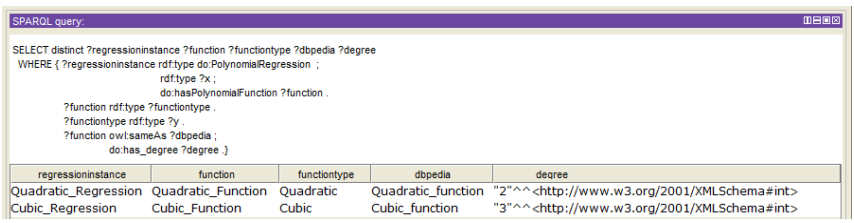


Figura E.36: Consulta sobre a função ajustada na regressão polinomial e seu tipo, grau relacionado e associação com a DBpedia.

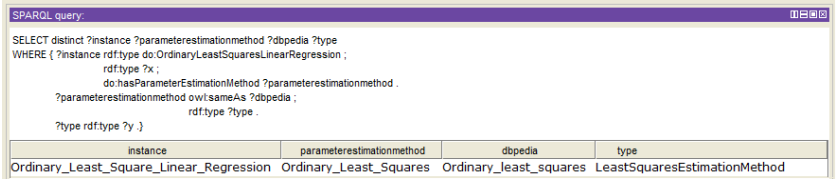


Figura E.37: Consulta sobre o método de estimação da regressão OLS, seu tipo e associação com a DBpedia.

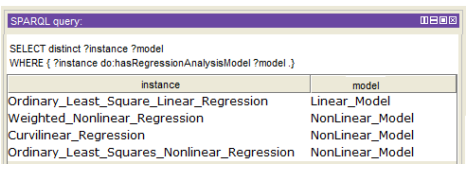


Figura E.38: Consulta sobre o modelo ajustado na regressão.

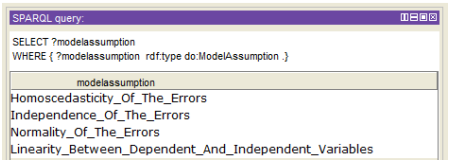


Figura E.39: Consulta sobre as suposições do modelo de regressão simples.

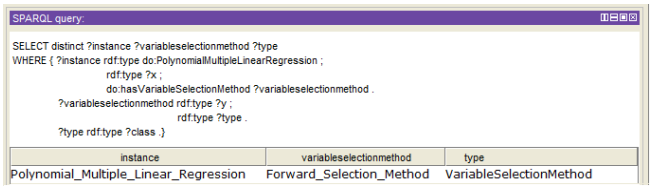


Figura E.40: Consulta sobre o método de seleção de variáveis independentes usado na análise de regressão múltipla.

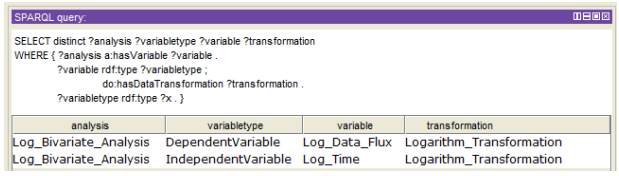


Figura E.41: Consulta sobre transformações feitas nas variáveis dependente e independente.

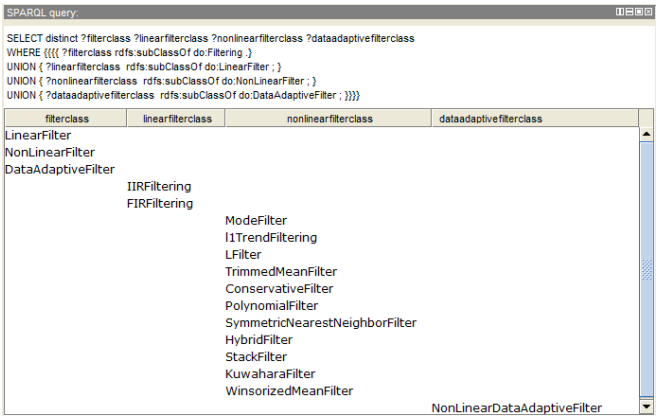


Figura E.42: Consulta sobre como são classificados os filtros.

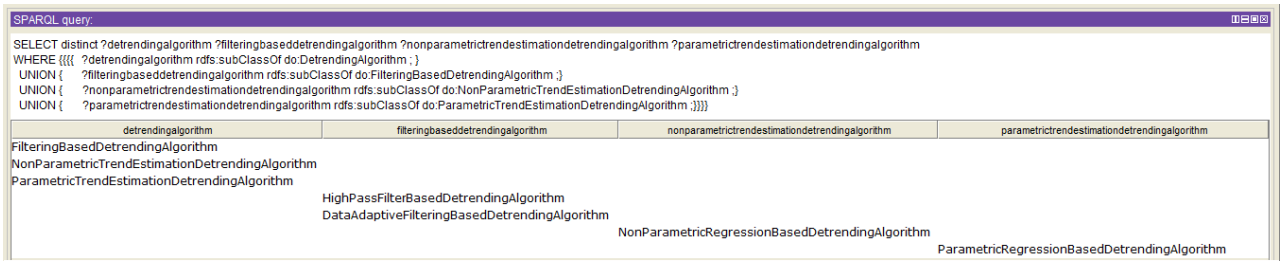


Figura E.43: Consulta sobre como são classificados os algoritmos/softwares conforme os métodos usados no passo de *detrending*.

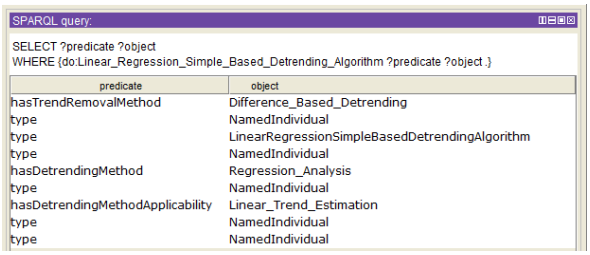


Figura E.44: Consulta sobre os relacionamentos do algoritmo de regressão linear simples.

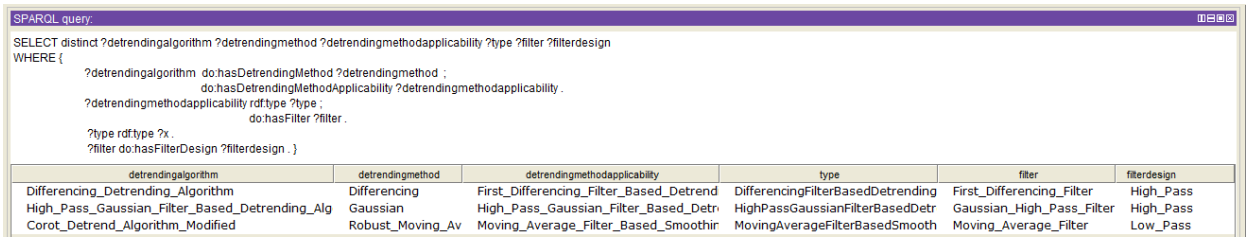


Figura E.45: Consulta sobre o método, sua aplicabilidade e o tipo dos filtros e seu *design* que tem aplicabilidade nos algoritmos de *detrending*.

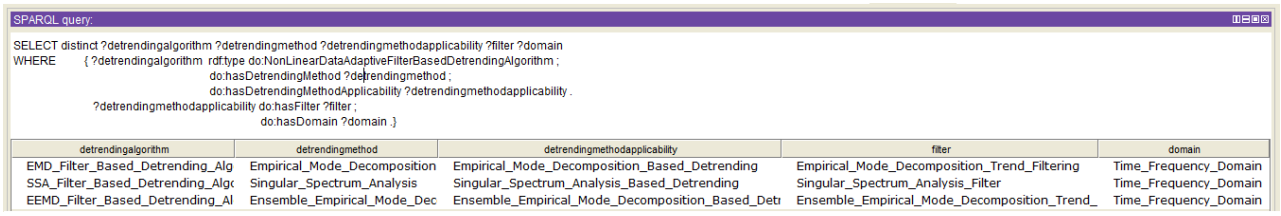


Figura E.46: Consulta sobre o algoritmo, o método e o domínio dos filtros adaptativos aos dados que tem aplicabilidade em *detrending*.

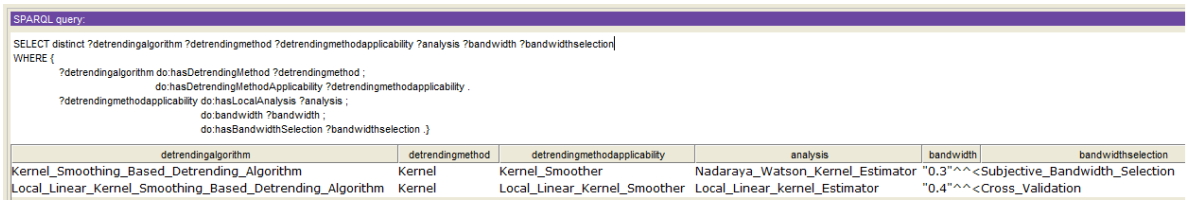


Figura E.47: Consulta sobre métodos e parâmetros usados para suavização kernel.

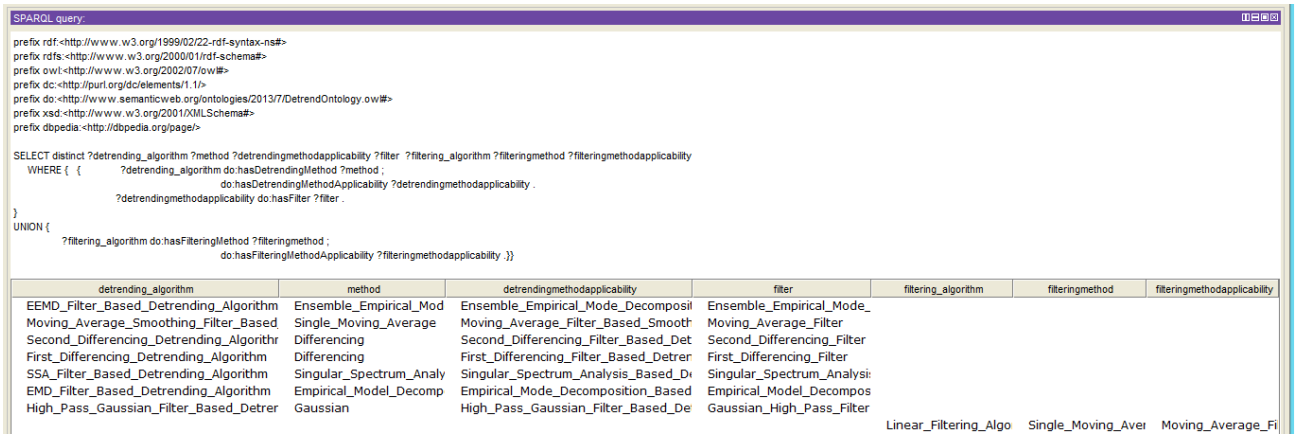


Figura E.48: Consulta sobre a aplicabilidade dos filtros em algoritmos de *detrending* e de filtragem de ruído.

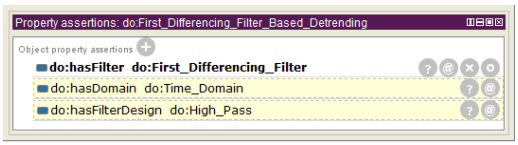


Figura E.49: Exemplo de inferência em *detrending* baseado em filtro passa alta frequência.

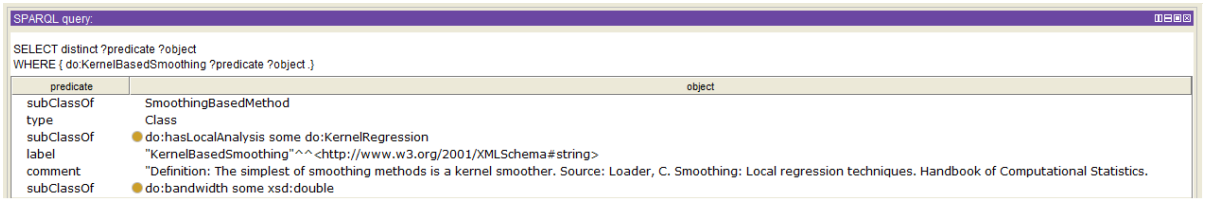
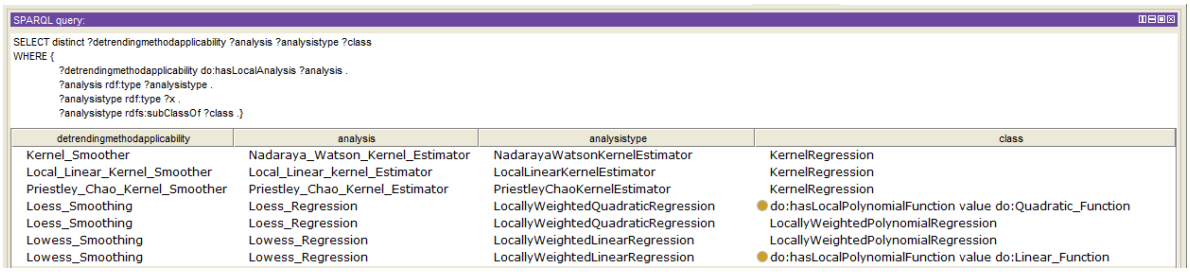


Figura E.50: Consulta sobre parâmetros do método de suavização kernel.

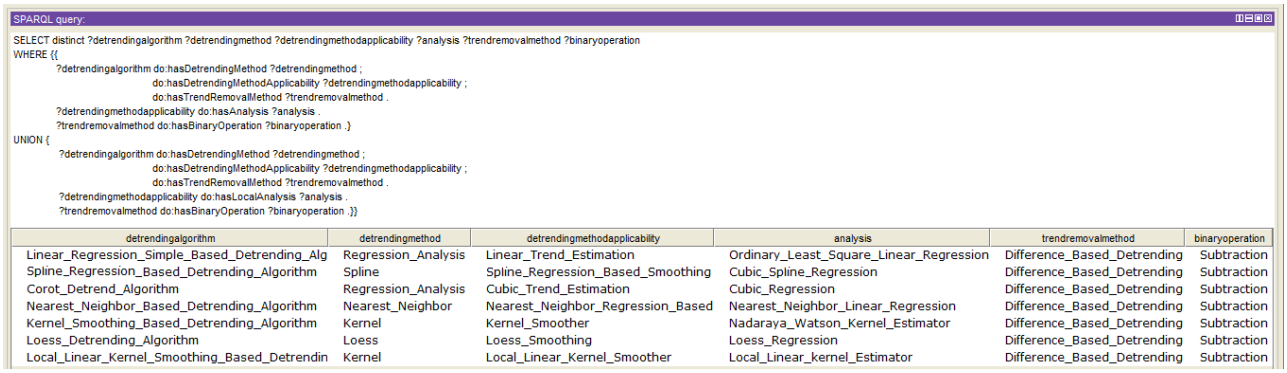


SPARQL query:

```
SELECT distinct ?detrendingmethodapplicability ?analysis ?analysistype ?class
WHERE {
        ?detrendingmethodapplicability do:hasLocalAnalysis ?analysis .
        ?analysis rdf:type ?analysistype .
        ?analysistype rdf:type ?x .
        ?analysistype rdfs:subClassOf ?class .}
```

detrendingmethodapplicability	analysis	analysistype	class
Kernel_Smoother	Nadaraya_Watson_Kernel_Estimator	NadarayaWatsonKernelEstimator	KernelRegression
Local_Linear_Kernel_Smoother	Local_Linear_kernel_Estimator	LocalLinearKernelEstimator	KernelRegression
Priestley_Chao_Kernel_Smoother	Priestley_Chao_Kernel_Estimator	PriestleyChaoKernelEstimator	KernelRegression
Loess_Smoothing	Loess_Regression	LocallyWeightedQuadraticRegression	do:hasLocalPolynomialFunction value do:Quadratic_Function
Loess_Smoothing	Loess_Regression	LocallyWeightedQuadraticRegression	LocallyWeightedPolynomialRegression
Lowess_Smoothing	Lowess_Regression	LocallyWeightedLinearRegression	LocallyWeightedPolynomialRegression
Lowess_Smoothing	Lowess_Regression	LocallyWeightedLinearRegression	do:hasLocalPolynomialFunction value do:Linear_Function

Figura E.51: Consulta sobre tipo de análise relacionada a métodos de suavização local, seu tipo e aplicabilidade.



SPARQL query:

```
SELECT distinct ?detrendingalgorithm ?detrendingmethod ?detrendingmethodapplicability ?analysis ?trendremovalmethod ?binaryoperation
WHERE {{
        ?detrendingalgorithm do:hasDetrendingMethod ?detrendingmethod ;
                             do:hasDetrendingMethodApplicability ?detrendingmethodapplicability ;
                             do:hasTrendRemovalMethod ?trendremovalmethod .
        ?detrendingmethodapplicability do:hasAnalysis ?analysis .
        ?trendremovalmethod do:hasBinaryOperation ?binaryoperation .}
UNION {
        ?detrendingalgorithm do:hasDetrendingMethod ?detrendingmethod ;
                             do:hasDetrendingMethodApplicability ?detrendingmethodapplicability ;
                             do:hasTrendRemovalMethod ?trendremovalmethod .
        ?detrendingmethodapplicability do:hasLocalAnalysis ?analysis .
        ?trendremovalmethod do:hasBinaryOperation ?binaryoperation .}}
```

detrendingalgorithm	detrendingmethod	detrendingmethodapplicability	analysis	trendremovalmethod	binaryoperation
Linear_Regression_Simple_Based_Detrending_Alg	Regression_Analysis	Linear_Trend_Estimation	Ordinary_Least_Square_Linear_Regression	Difference_Based_Detrending	Subtraction
Spline_Regression_Based_Detrending_Algorithm	Spline	Spline_Regression_Based_Smoothing	Cubic_Spline_Regression	Difference_Based_Detrending	Subtraction
Corot_Detrend_Algorithm	Regression_Analysis	Cubic_Trend_Estimation	Cubic_Regression	Difference_Based_Detrending	Subtraction
Nearest_Neighbor_Based_Detrending_Algorithm	Nearest_Neighbor	Nearest_Neighbor_Regression_Based	Nearest_Neighbor_Linear_Regression	Difference_Based_Detrending	Subtraction
Kernel_Smoothing_Based_Detrending_Algorithm	Kernel	Kernel_Smoother	Nadaraya_Watson_Kernel_Estimator	Difference_Based_Detrending	Subtraction
Loess_Detrending_Algorithm	Loess	Loess_Smoothing	Loess_Regression	Difference_Based_Detrending	Subtraction
Local_Linear_Kernel_Smoothing_Based_Detrendin	Kernel	Local_Linear_Kernel_Smoother	Local_Linear_kernel_Estimator	Difference_Based_Detrending	Subtraction

Figura E.52: Consulta sobre quais métodos têm aplicabilidade em regressão, algoritmos relacionados e parâmetros.